Shoemaker

R-977

INERTIAL NAVIGATION SYSTEM STANDARDIZED SOFTWARE DEVELOPMENT

FINAL TECHNICAL REPORT

Volume II of IV

INS SURVEY AND ANALYTICAL DEVELOPMENT

June 1976

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER R-977	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INERTIAL NAVIGATION SYSTEM STANDARDIZED SOFTWARE DEVELOPMENT, FINAL TECHNICAL REPORT, VOLUME II , INS SURVEY AND ANALYTICAL DEVELOPMENT		5. TYPE OF REPORT & PERIOD COVERED 3/15/76-4/30/76 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Sciegienny, R. Nurse, J. Wexler, P. Kampion		8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-1149
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. 68 Albany Street Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task 4.2.1
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory Wright-Patterson Air Force Base Dayton, Ohio 45433		12. REPORT DATE June 1976 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Inertial Navigation
Computer Algorithms
Computation Errors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Section I of this volume provides a summary of the navigation computations for seven aircraft inertial navigation systems (INS): four of the INS are local vertical systems (north slaved, wander azimuth and free azimuth); one INS is space stable; and the remaining two are strapdown.

Section 2 presents the detailed navigation and attitude equations, applicable to any of the aforementioned INS, using both local vertical wander azimuth (LVWA) and space stable computational frames. The LVWA frame is selected for

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (continued)

the development of "standard" navigation algorithms using a standardized notation and symbology. The "baseline" algorithm is similar to that used in several existing moderate accuracy, local vertical, aircraft INS.

The "upgraded" algorithm contains several minor modifications to the "baseline" algorithm which improve the computational accuracy of the latter. The modifications are derived in the appendices. Transformations required to use the "standard" algorithms with space stable or strapdown systems are also presented in this section.

Section 3 summarizes the INS computations presented in the previous section.

Section 4 provides a survey of the use of INS computations by other avionics subsystems, with particular reference to the parameters, formats, data rates, and accuracy requirements.

Section 5 presents the tradeoffs leading to the selection of the LVWA computational frame for the "standard" algorithm.

Section 6 (and Appendix A) present the standardized symbology and frame definition employed in the "standard" algorithms and simulator development.

R-977

INERTIAL NAVIGATION SYSTEM STANDARDIZED SOFTWARE DEVELOPMENT

FINAL TECHNICAL REPORT

VOLUME II

INS Survey

and

Analytical Development

June 1976

Approved: W.G. Denhard
W.G. Denhard

The Charles Stark Draper Laboratory, Inc.

Cambridge, Massachusetts 02139

ACKNOWLEDGEMENT

This four-volume report was prepared under USAF Contract F33615-75-C-1149 by Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts, in accordance with Section 4 of the contract. The monitoring Air Force project engineer is Captain E. Harrington (RWA-2), Air Force Avionics Laboratory, Dayton, Ohio.

The Draper Laboratory Program Manager for this task is Dr. George T. Schmidt and the Lead Engineer is Arthur Ciccolo. The coordinator of this report is Janusz Sciegienny and the authors are Janusz Sciegienny, Roy Nurse, Peter Kampion and John Wexler.

The authors express their appreciation to Mr. W. Shephard, Mr. D. Kaiser and Mr. Stan Musick of the Air Force Avionics Laboratory for their assistance during the course of this contract.

TABLE OF CONTENTS

VOLUME II

TABLE OF CONTENTS (CONTINUED)

VOLUME II

TABLE OF CONTENTS (CONTINUED)

VOLUME II

TABLE OF CONTENTS (CONTINUED)

VOLUME II

APPENDIX

VOLUME II

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

VOLUME II

LIST OF TABLES

VOLUME II

1.0 SURVEY OF INS COMPUTATIONS

1.1 Introduction

A survey of INS navigation computations was initiated to investigate the mechanization equations used in representative types of aircraft-INS configurations either deployed or under development at this time.

The survey results were used in design of the computer error simulation program, described in Volume III of this report.

The survey included INS with a local-level stabilized platform, with a space-stabilized platform, and with strapdown configurations. The surveyed INS used either the local-level or the space-stabilized computational frame during the INS navigation computations. Table 1-1 contains a list of the surveyed INS, the type of platform stabilization, and the type of computational frame used during the navigation computations.

TABLE 1-1. List of Surveyed INS

INS	TYPE OF PLATFORM STABILIZATION	NAVIGATION COMPUTATIONAL FRAME USED
KT-73	Local-level geodetic	North-pointing below 70° latitude Wander-azimuth above 70° latitude
LN-15	Local-level wander-azimuth	Local-level geodetic wander-azimuth
H-386	Local-level free-azimuth	Local-level free-azimuth geocentric
Carousel	Local-level free-azimuth rotating-level gyros and accelerometers	Local-level geodetic free-azimuth
GEANS	Space-stabilized	Space-stabilized
SIGN III	Strapdown	Local-level geocentric
SIRU	Strapdown	Local-level geodetic

The following computational characteristics were investigated:

a. Commonality in navigation computations mechanization among various INS types.

b. Form of gravity models employed in the navigation computations.

c. Detailed form of the navigation equations mechanizations.

d. Commonality of coordinate frames and coordinate transformations symbology employed in the navigation equations.

The survey results are presented in sections 1.3 to 1.8 and summarized in section 1.2. The survey results were also used to generate sections 2, 3, 5 and 6.

1.2 Summary and Conclusions

1.2.1 Commonalities and Differences in Navigation Computations

- A comparison of the commonalities and differences in the navigation-computation format of the surveyed INS with local-level mechanization appears in Table 1-2.

TABLE 1-2. Commonalities and Differences in the Navigation-Computation Format of the Surveyed INS with Local-Level Mechanization

COMMONALITIES	DIFFERENCES
• The navigation computations are performed in a local-level computational frame with world-wide navigation capabilies. • A first-order direction-cosine transformation matrix from the platform frame to the earth-fixed frame is used in position computations. • Barometric altimeter data is used to bound the vertical-channel computations.	• The local-level computational frames used in mechanization of the INS navigation equations are: (1) LN-15 and KT-73 (above 70° latitude) INS use geodetic wander-azimuth computational frame. (2) KT-73 (below 70° latitude) INS uses geodetic north-pointing computational frame. (3) H-386 INS uses the geocentric free-azimuth computational frame. (4) Carousel INS uses geodetic free-azimuth computational frame.

TABLE 1-2. Commonalities and Differences in the Navigation-Computation Format of the Surveyed INS with Local-Level Mechanization (Continued)

COMMONALITIES	DIFFERENCES
	• The direction-cosine matrix, representing the transformation from the computational frame to the earth-fixed frame, used in the KT-73, LN-15, and Carousel INS computations is updated using the angular velocity of the computational frame with respect to the earth-fixed frame. In the H-386 INS computations, the updated direction-cosine matrix is from the computational frame to the inertial frame. This matrix is updated using the level-axis components of the angular velocity of the computational frame with respect to the inertial frame. The earth's angular velocity is then used to update the matrix from the inertial frame to the earth-fixed frame. • All nine elements of the direction-cosine matrix used in the KT-73, H-386, and Carousel INS computations are updated. In the LN-15 INS computations, only six direction-cosine matrix elements are updated. • Orthonormalization of the direction-cosine matrix is used only in the H-386 INS computations. • In the vertical-channel navigation computations, the barometric altimeter data is combined with the inertial data as follows: (1) KT-73 and LN-15 INS computations use third-order damping loops. (2) Carousel INS computations use second-order damping loops. (3) H-386 INS uses barometric data directly (no inertial vertical computations are mechanized).

- The surveyed INS with a space-stabilized platform (GEANS), used a space-stablilized computational frame in the mechanization of the navigation computations. The inertially derived data is combined with the barometric altimeter data on the acceleration and the velocity levels (second-order baro-inertial loop in the inertial frame).

- The commonalities and differences in the navigation-computation format of the surveyed strapdown INS appear in Table 1-3.

TABLE 1-3. Commonalities and Differences in the Navigation-Computation Format of the Surveyed Strapdown INS

COMMONALITIES	DIFFERENCES
• The navigation computations are performed in a local-level north-pointing computational frame. The INS computation mechanization is therefore not suitable for operation near the earth's poles. • A quaternion is used to express the coordinate transformation from the body frame to the computational frame. • Position is computed from the integration of velocity.	• The SIGN III INS uses the geocentric computational frame, while the SIRU INS uses the geodetic computational frame. • The SIGN III INS uses barometric data in the navigation computations, while SIRU INS uses constant altitude in the navigation computations.

1.2.2 Comparison of Gravity Models

The gravity models used in the INS navigation computations are based on the ellipsoidal gravity formula[1]. The formula expresses the average geodetic vertical component of the earth's gravity at sea level in the form of a Chebychev expansion approximation. The gravity equation consists of a constant term and terms proportional both to the square and to the fourth powers of the sine of latitude. Above sea level, the gravity model contains an additional vertical component proportional to the altitude, and a level (north-south) term proportional to the product of the altitude and the sine of twice the latitude angle. The magnitude of each of these additional terms is less than 24μg.

The gravity models used in the KT-73, LN-15, and Carousel navigation computations consist of the geodetic vertical components. The gravity model contains the constant term, the term proportional to the sine of the latitude squared, and the term proportional to the altitude. Omission of the fourth-power sine of the latitude term results in a gravity error of less than 23μg. The KT-73 or LN-15 gravity models use the same values of the model terms. The Carousel gravity model term values are slightly lower than the corresponding term values in the KT-73 or LN-15 gravity models. The difference is due to a slightly different value of the earth parameters used in computing the model term values. The resultant difference between the gravity magnitude in the KT-73 or LN-15 and the Carousel computations is less than 20μg.

The gravity models used in the H-386 and SIGN III INS navigation computations consist of a geocentric-vertical and a geocentric-north component. The gravity components are computed with an accuracy on the order of 20μg.

The gravity model used in GEANS navigation computations consists of gravity terms computed in inertial rectangular coordinates. The gravity is computed with an accuracy of the order of 0.1μg.

The magnitude of the gravity error resulting from the gravity model approximations used in KT-73, LN-15, H-386, Carousel, and SIGN III INS navigation computations is comparable with the magnitudes of the gravity anomalies. In addition, the vertical component of gravity error has not significant effect on the navigation errors in navigation computations supplemented by the barometric altimeter data. The horizontal component error does, however, introduce navigation errors.

1.2.3 Detailed Form of Navigation-Equations Mechanization

As part of the INS navigation-computation survey, a detailed study of navigation-equations implementation was investigated.

Three types of INS configurations (local-level, space-stabilized, and strapdown) and two types of computational frames (local-level and space-stabilized) were included in this investigation. The resultant six forms of the navigation-computation mechanization are represented by detailed flow diagrams. In addition, specialized aspects of the computations used in a strapdown INS were investigated to determine the most efficient mechanization of the strapdown navigation equations in a local-level computational frame. The details of the navigation equations mechanization are presented in section 2 of this volume.

1.2.4 Commonality of Symbology

The survey of the various INS navigation computations indicates a lack of standardization in symbology of frames and transformations. The symbology used varies not only among various INS types, but it is also not always consistent in the same type of INS computations.

In the KT-73 INS computations, for example, the local-level platform and the computational frame use different definitions of axes' orientation. Also, the LN-15 series INS do not use consistent symbology of frame definitions. The survey results, presented in section 1, employ symbology used in the surveyed reports.

A proposed standardized notation based on the Honeywell convention used in SPN/GEANS [2] is presented in section 6 and it is used in sections 2 and 3 of this volume.

1.3 KT-73 Inertial Navigation Computations

1.3.1 General

The KT-73 inertial measurement unit (IMU), manufactured by the Kearfott Division of Singer, is used in the AN/ASN-90(V) inertial measurement set (IMS). The IMS is a part of the inertial navigation system (INS) used on the Air Force A-7D tactical fighter bomber, the Air Force AC-130E gunship and the Navy A-7E attack fighter-bomber.

The INS uses the AN/ASN-91(V) computer (NWDC), manufactured by IBM, to provide navigation and weapon delivery information to other aircraft avionic systems. The NWDC receives inputs from many avionic systems including the AN/APN-90(V) doppler radar set, manufactured by Kearfott, and the CP-953A/AJQ air data computer, manufactured by Air Research. The NWDC is a 16,000-word, 16-bit, fixed-point, magnetic-core-memory digital computer.

The KT-73 INS contains a four-gimbal platform. The platform employs the two-degree-of-freedom gyroflex gyros, with one-degree-of-freedom caged. Torque-restrained, pendulum-type accelerometers mounted on the platform are used to measure the specific force. The horizontal components of the specific force are measured by a two-axis accelerometer and the vertical specific force is measured by a single-axis accelerometer.

The platform stabilization below 70° latitude is local-level, north-pointing and above 70° latitude the stabilization becomes local-level, wander azimuth. The vertical channel computations are stabilized by the barometric altimeter data. The stabilization mechanization employs a constant-gain, third-order loop. The doppler radar measured velocity is used for damping of Schuler oscillations in the inertially derived velocity. The velocity damping mechanization employs second-order constant gain loops.

This section contains a description of the navigation computations of the position, the velocity and the attitude employed in the pure inertial mode of operation (with no doppler-radar measurements). The description is based on reports generated at the Naval Weapon Center, China Lake, California [3]. The Center is responsible for the maintenance, validation, and modification of the A-7 D/E navigational and the weapons delivery equations.

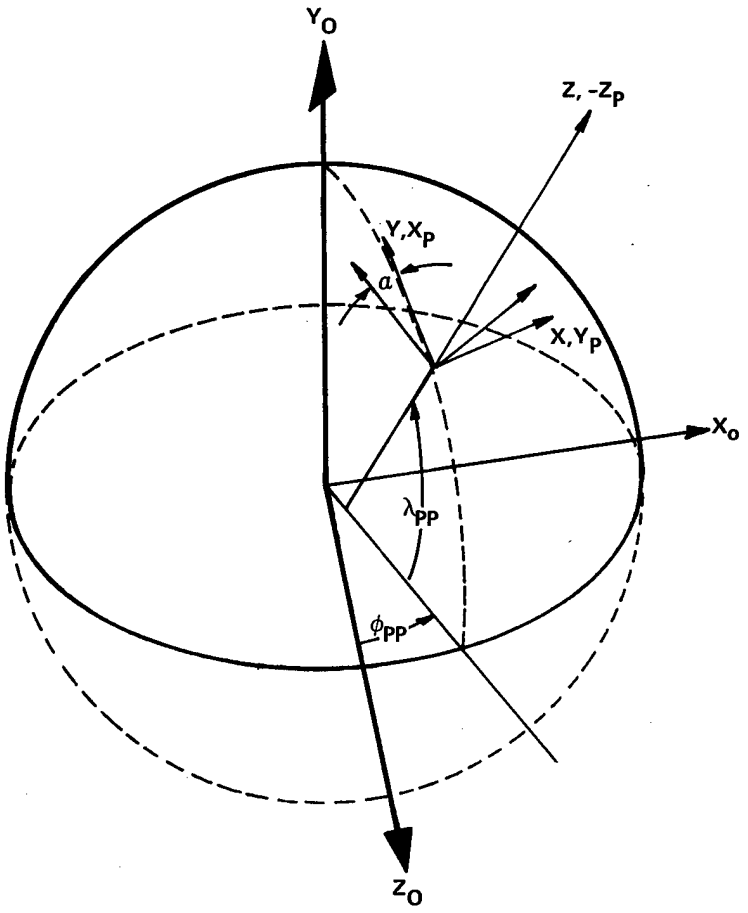
1.3.2 Coordinate Frames

This section contains a definition of the frames used during the navigation computations of the position, the velocity and the attitude. The symbology of frames and transformations used in this section is obtained from Reference (3).

The frames used during the navigation computations of the position and the velocity are the earth centered frame, x_o, y_o, z_o, the platform reference frame, x_p, y_p, z_p, and the computational frame, x, y, z, as shown in Figure 1-1. The computational frame coincides with the local vertical geodetic frame for latitude $L < 70°$. For $L > 70°$ the computational frame becomes a wander azimuth frame. (The wander angle is equal to α.)

The platform frame axes are also in a local geodetic frame north-point for $L < 70°$ latitude, and wander-azimuth for $L > 70°$. The platform axes do not coincide with the corresponding axes of the computational frames. There seems to be no justification for choosing a different definition for the platform and the computational axes.

The aircraft body axes. the gimbal axes and the definition of the attitude angles are shown in Figure 1-2.



X_O, Y_O, Z_O = EARTH FIXED REFERENCE
X,Y,Z = LOCAL GEODETIC, EAST, NORTH, UP
X_P, Y_P, Z_P = PLATFORM REFERENCE, NORTH, EAST, DOWN
λ_{PP} = LATITUDE, ϕ_{PP} = LONGITUDE, α = WANDER ANGLE

Figure 1-1. Definition of KT-73 Navigation Axes Frames

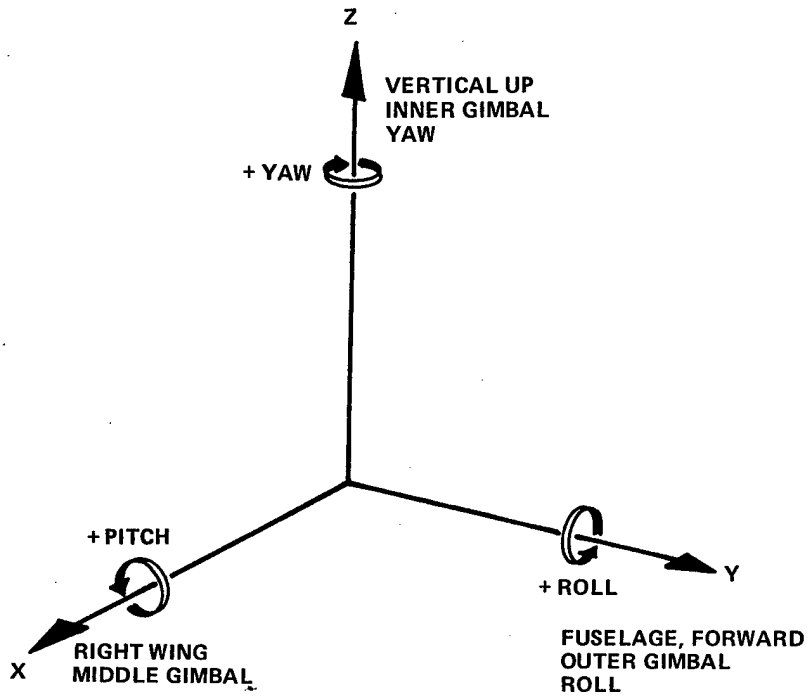


Figure 1-2. Aircraft and KT-73 Gimbal Axes

1.3.3 Position and Velocity Computations

- The KT-73 inertial navigation computations are performed in a local-vertical, geodetic frame, north-pointing below 70° latitude and with wander-azimuth above 70° latitude.

- The duration of a basic iterative cycle is 40 milliseconds.

- The navigation computations use direction cosine matrix C, in mechanization of transformation from the computational frame to the earth fixed frame. The matrix is updated using a first-order approximate solution of the matrix equation:

$$\dot{C} = C\phi$$

  where ϕ is the skew symmetric matrix representing the angular velocity of the computational frame with respect to the earth-fixed frame expressed in the computational frame coordinates. The approximation is solved using double-precision computation.

- The vertical channel computations employ a third-order damping system. (The difference between the inertially derived altitude and the barometric altitude is applied through the proportional and the integral channels to the inertially sensed acceleration. The altitude difference is also applied to the inertially derived velocity.)

- The gravity computations employ the gravity model represented by the following equation:

$$g = 32.0882\ (1 + 5.29 \times 10^{-3} \sin^2\lambda_{pp} - 10^{-7} h_B)\ \text{ft/sec}^2$$

  where λ_{pp} is the geodetic latitude and h_B is the barometric altitude.

Figure 1-3 presents a functional flow diagram of the KT-73 INS computations of position and velocity in the local level, geodetic, wander azimuth computational frame. (For a north-pointing computational frame the wander angle α is zero.) The velocity increments $\Delta\bar{V}$ sensed by the accelerometers and resolved along the computational frame axes are combined with the V/R terms and with the Coriolis correction terms. The computed horizontal velocity is resolved through the wander angle α, into the east and north components. Computation of the vertical velocity component, V_Z, requires the magnitude of the local gravity. In addition, the barometric altitude, h_B, is used in a third-order damping loop to bound the vertical channel instability. The computed horizontal velocity is used in the computation of the angular rate, ϕ, of the computational frame with respect to the earth-fixed frame. The ϕ is used to update the direction cosine matrix transformation between the computational frames and the earth fixed frame. The updating is based on solution of the differential equation $\dot{C} = C\phi$. The latitude, λ_{pp}, longitude, ϕ_{pp}, and the wander angle, α, are computed from the C matrix terms. The components of the angular rate ϕ and the matrix C are used in computation of the gyro torquing signals ω_x, ω_y, ω_z.

1.3.4 Attitude Computations

The aircraft attitude is defined by the orientation of the aircraft body axes shown in Figure 1-2, with respect to a local geodetic axes shown in Figure 1-1. The attitude angles, roll, pitch, yaw, are defined respectively by the angles of rotation about the x,y,z aircraft axes required to bring the aircraft frame axes to coincide with the corresponding local geodetic frame axes.

Figure 1-4 presents a functional flow diagram of the KT-73 navigation computation of attitude in the local-level, geodetic, wander azimuth computational frame. The aircraft attitude angles are computed by solving the following matrix equation.

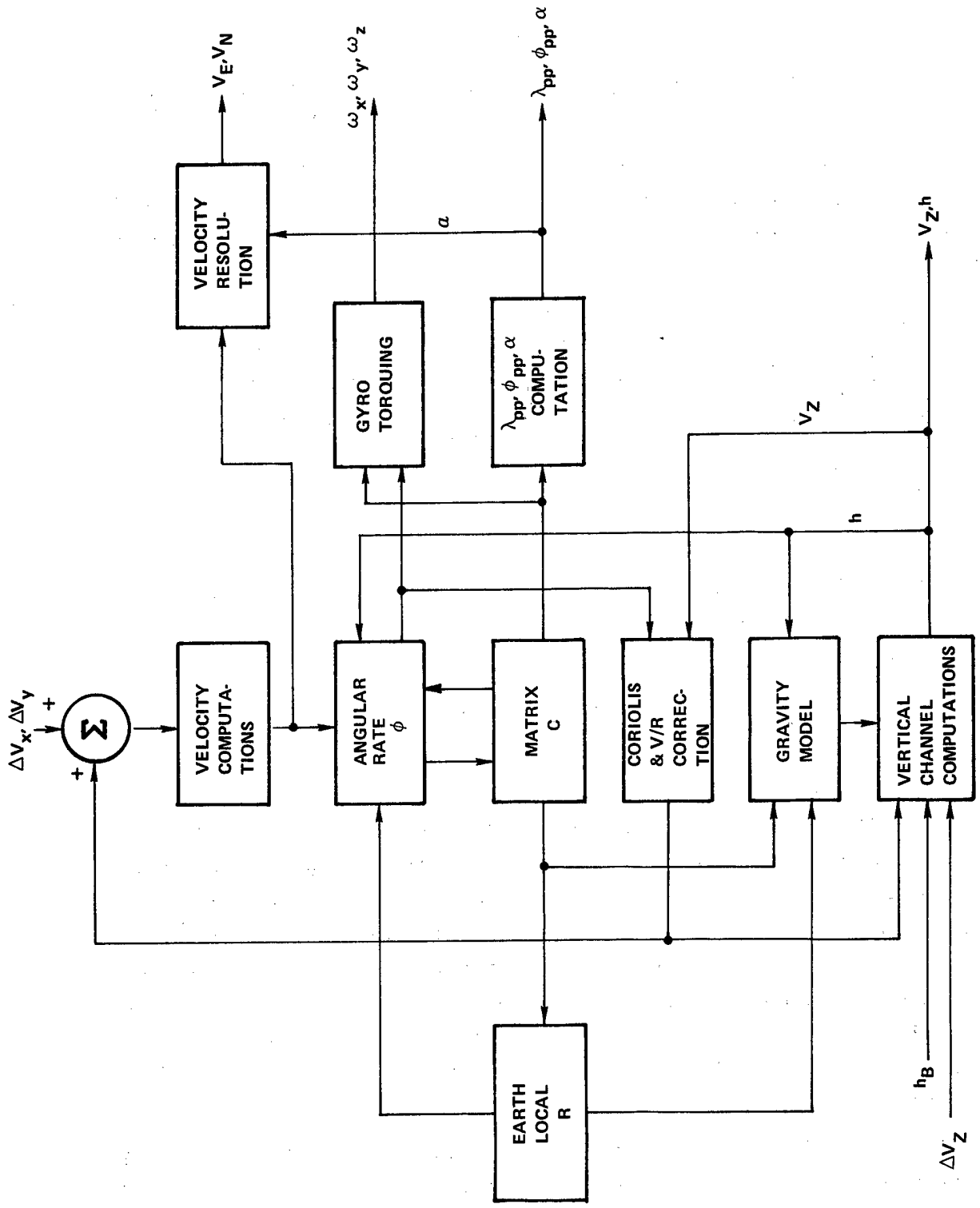


Figure 1-3. KT-73 Navigation Computations of Position and Velocity Functional Flow Diagram

Attitude angles matrix = Transformation matrix from the aircraft body axes to the local geodetic axes

The transformation matrix from the body axes to the geodetic axes is a function of the four gimbal angles and the wander angle.

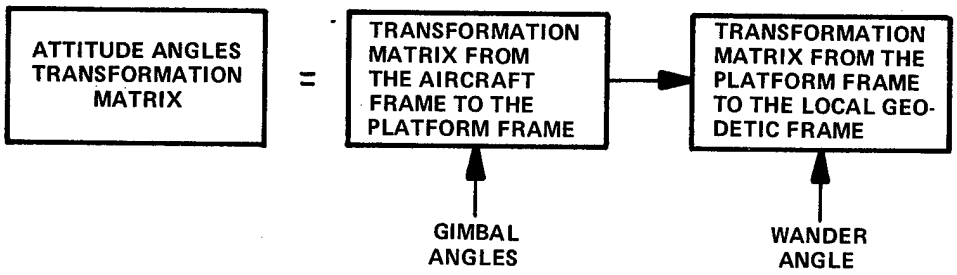


Figure 1-4. KT-73 Navigation Computations of Attitude - Functional Block Diagram

1.4 LN-15 Inertial Navigation Computations

1.4.1 General

The LN-15 series INS, manufactured by Litton, is used in various military and civilian aircraft including high performance fighter-bomber aircraft. The IMU contains a four-gimbal platform series P-500. The platform employs two, two-degree-of-freedom gyros, series G300G2, with one-degree-of-freedom of one gyro caged. Three torque-restrained, pendulum-type accelerometers, series A200D, mounted on the platform are used to measure the specific force. Most of the LN-15 series INS employ the LC-72B computer.

Definition of the axes frame used in the navigation computations is not the same for all the LN-15 series INS. This section presents the LN-15 navigation computations described in an Intermetrics, Inc. report[4].

1.4.2 Coordinate Frames

The frames used during the navigation computations[4] of the position and the velocity, shown in Figure 1-5, are the earth centered frame, X, Y, Z, and the local vertical geodetic wander-azimuth frame, x, y, x. The x, y, z frame coincides with the computational frame and with the platform reference frame. Note that the LN-15 INS and the KT-73 INS navigation computations use the same definition of the earth centered and computational frames.

Frame definition is dependent on the software implementation. For the LN-15S used on B-1 aircraft the frames are defined as follows by Boeing[5].

The earth centered frame has the x axis passing through the Greenwich Meridian and the z axis along the earth axis of rotation. The x axis is positive through the Greenwich Meridian and the z axis is positive through the North Pole. The platform reference frame, coincident with the computational frame has the z axis along the local gravity vector and the angle between x axis and the North is the wander angle, α. The positive direction of the z axis is up and the positive direction of the x axis for $\alpha = 0$, is North. The wander angle, α, is defined to be positive counterclockwise from North.

1.4.3 Position and Velocity Computations

- The LN-15 navigation computations are performed in a local-vertical, geodetic, wander-azimuth frame.
- The duration of a basic iterative cycle is 1/16 second.
- The navigation computations use a direction cosine matrix, C, representing the coordinate transformation from the computational frame, to the earth-fixed frame. Only six matrix terms (out of nine total) are computed.
- The vertical channel computations employ a third-order damping system.

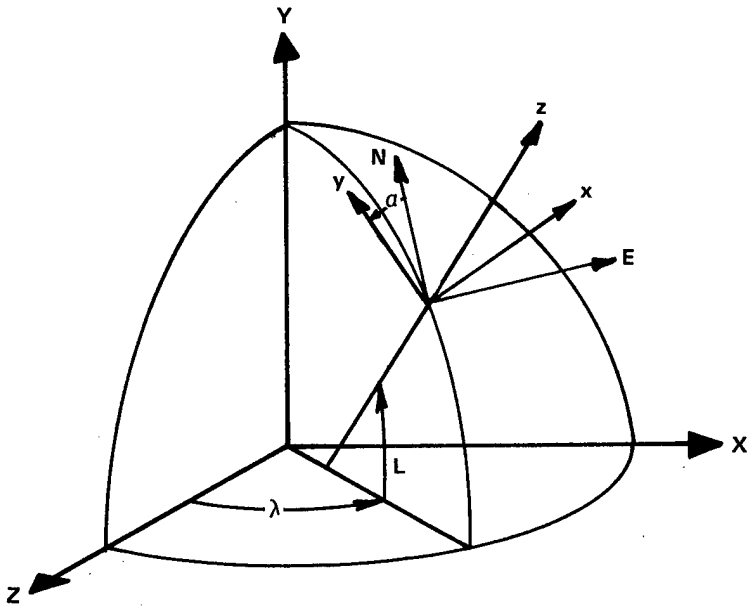


Figure 1-5. LN-12 Definition of Coordinate Frames

- The gravity computations in the gravity model are represented by the following equation.

$$g_z = -32.0881576\ (1 + 5.2888 \times 10^{-3} \sin^2 L - 0.955 \times 10^{-7} h)\ ft/sec^2$$

where L is the geodetic latitude and h is the altitude computed in the vertical channel.

The navigation computations of position and velocity are represented by the KT-73 flow diagram shown in Figure 1-3. Note that the KT-73 and the LN-15 INS navigation computations are represented by the same functional flow diagram, but the computations employ different symbology.

1.4.4 Attitude Computations

The LN-15 attitude computations have the same form as the KT-73 attitude computations described in section 1.3.4. Definition of the axes frames used in the LN-15 attitude computations is, however, not the same as in the KT-73 attitude computations.

1.5 H-386 Inertial Navigation Computations

1.5.1 General

H-386 is a designation for a member of a family of Honeywell-designed and built aircraft inertial navigation systems (INS). Some applications are classified; others are for aided system demonstrations including star tracker, satellite and Doppler aiding. The basic INS consists of the inertial measureing unit (IMU), the inertial electronics unit (IEU), the control and display unit (CDU) and the digital computer.

A Honeywell Mark III computer provides the navigation and steering information, plus discrete and analog outputs, to other aircraft avionics systems. The prime inputs from other avionics systems are barometric altitude, reference position and Greenwich mean time (GMT). The computer is a 24-bit, fixed-point, drum memory unit.

The IMU is a three-gimbal platform, employing three-single-degree-of-freedom, gas bearing, rate-integrating gyros (GG159) and three single axis, flexipivot, hinged pendulum accelerometers (GG177). Both gyros and accelerometers are pulse-rebalanced providing direct digital outputs.

The platform stabilization is local-geocentric vertical, free-azimuth. A north-slaved azimuth mode is available for fixed base testing. A two accelerometer mechanization is employed-altitude being supplied by the barometric altimeter.

This section describes the navigation computations of position, velocity and wander angle in the two accelerometer, barometric altitude, pure-inertial mode based on internal Honeywell documentation[6], [7].

1.5.2 Coordinate Frames

Several coordinate frames are employed in the H-386 navigation system mechanization; these are the platform coordinates, the geocentric latitude and longitude frame, the geodetic latitude and longitude frame, and an earth-fixed, rectangular coordinate frame with the origin at the center of the earth.

The platform coordinate frame has its origin at the center of the optical cube on the azimuth block. The x, y, and z axes are defined as the normals to the theoretical cube faces. This coordinate frame is maintained so that the z axis is pointed toward the geocentric center of the earth. The x axis is related to north by a rotation about z through the angle, ψ_p, as shown in Figure 1-6. Acceleration measurement and gyro torquing are done in this coordinate frame. INS velocity is also computed in this coordinate frame.

The geocentric latitude-longitude frame is related to the xyz frame by the angle, ψ_p. The INS position is computed in this coordiate frame. The position is then expressed in the geodetic latitude-longitude frame. The geodetic frame is related to the geocentric latitude-longitude frame by a rotation about the east/west axis of θ_{GC}, as shown in Figure 1-7.

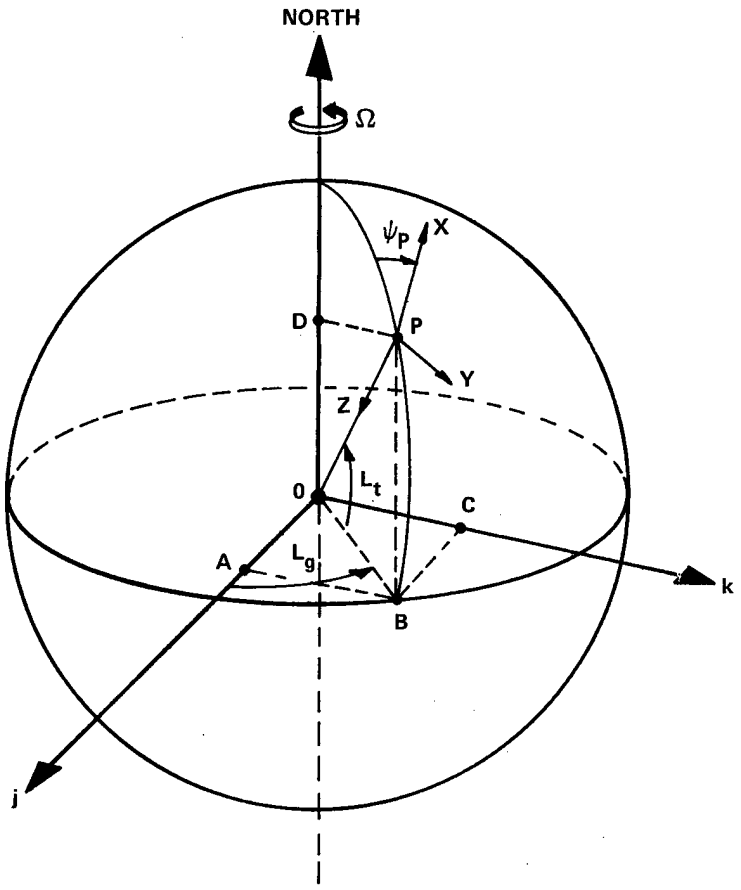


Figure 1-6. Definition of H-386 Navigation Axes Frame

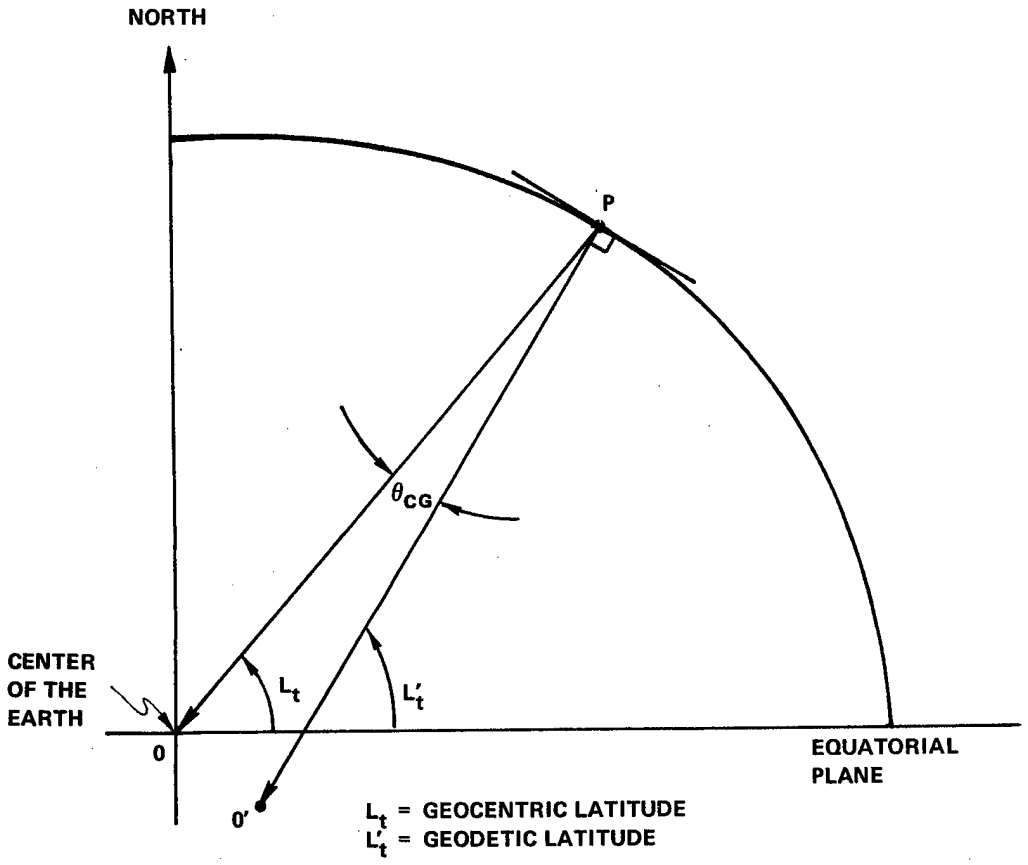


Figure 1-7. Relationship Between Geocentric and Geodetic Latitudes

For computations of steering and course selection, it is convenient to express the INS position in terms of the direction cosines of the vector from the center of the earth to present position. In Figure 1-7, OP is this vector. The coordiante frame in which these direction cosines are expressed has its origin at the center of the earth, the i axis is directed along the earth's rotational axis, the j axis lies in the equatorial plane at longitude = 0 (the Greenwich Meridian) and the k axis lies in the equatorial plane at longitude = E90°. The direction cosines of OP in this frame are:

$$\lambda_1 = O \sin L_t$$

$$\lambda_2 = O \cos L_t \cos L_g$$

$$\lambda_3 = O \cos L_t \sin L_g$$

where L_t is the geocentric latitude and L_g is the longitude. Relations between the geodetic latitude, L_t' and the geocentric latitude, L_t, shown in Figure 1-7, are given by:

$$L_t = L_t' - K_{GC} \sin 2L_t'$$

where

$$K_{GC} = 3.367005 \times 10^{-3} \text{ rad} = 11.575 \text{ arc-minutes}$$

1.5.3 Position and Velocity Computations

Two mechanizations for the velocity and position computations have been employed with this system.

In the first (and earlier) mechanization the platform is stabilized in geocentric-vertical, free-azimuth frame (no torque other than drift compensation, is applied to the azimuth axis gyro). The inertial angular velocities of the platform level axes, ω_{IOx}, ω_{IOy}, are resolved through the wander angle, ψ_p, between the platform x axis and the north direction, into the

components along the geocentric, local-vertical, north-pointing, computational frame. The latitude and the longitude rates $\dot{L}_t$, $\dot{L}_g$ are then computed and integrated.

The problem of dividing the computed term $(\Omega + \dot{L}_g) \cos L_t$ by $\cos L_t$ is recognized by carrying $1/\cos L_t$ and the resulting $(\Omega + \dot{L}_g)$ in double precision. If computations determine that the INS is passing exactly over one of the poles an overflow will occur.

The wander angle, ψ_p, is computed from

$$\psi_{p_n} = \psi_{p_{n-1}} + \left[\omega_e \Delta t + \frac{\dot{L}_{g_n} + \dot{L}_{g_{n-1}}}{2} \Delta t \right] \sin L_t$$

The geocentric latitude, L_t, is converted to geodetic latitude, L_t', only for display.

The second (and later) mechanization computes ω_{IOx} and ω_{IOy} the same as the first mechanization, but then defines the platform-to-inertial reference frame transformation by a matrix A consisting of elements a_{ij}.

Position information, L_t, L_g, and ψ_p, is used in computation of A, which is updated by integrating $\dot{A} = A\Omega$, where Ω is the skew symmetrix matrix form of $\bar{\omega}_{IO}$, and $\omega_{IO} = \{\omega_{IOx}, \omega_{IOy}, 0\}$. A first order direction cosine update is employed to form

$$A_n = A_{n-1} + A_{n-1} \, \Omega \Delta t$$

Direction cosines of the position vector relative to the center of the earth are given by

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = Z(\omega_e t) \begin{bmatrix} -a_{13} \\ -a_{23} \\ -a_{33} \end{bmatrix}$$

L_t, L_g and ψ_p are obtained from the appropriate inverse trigonometric functions, viz:

$$L_t = \tan^{-1}\left(\lambda_1/\sqrt{\lambda_2^2 + \lambda_3^2}\right)$$

$$L_g = \tan^{-1}\left(\lambda_3/\lambda_2\right)$$

$$\psi_p = \tan^{-1}\left(-a_{12}/a_{11}\right)$$

The geocentric level components of earth relative velocity, V_E and V_N, are

$$V_E = R\left(\omega_{IOx}\cos\psi_p - \omega_{IOy}\sin\psi_p - \omega_e \cos L_t\right)$$

$$V_N = R\left(\omega_{IOy}\sin\psi_p + \omega_{IOy}\cos\psi_p\right)$$

where

ω_{IOx}, ω_{IOy} are the level components of the vehicle angular rates with respect to the inertial frame expressed in the platform coordinates. For wander angle, $\psi_p = 0$, the x platform axis is pointing north and the y platform axis is pointing east.

The "gravity" model, geocentric north and radial components, g_N and g_Z, are

$$g_N = -0.010489 \sin L_t \cos L_t \qquad (\text{ft/sec})^2$$

$$g_Z = 32.199497\left(1 + 1.847 \times 10^{-3}\sin^2 L_t - 0.262 \times 10^{-7} h_a\right) - 20925738\left(\omega_{IOx}^2 + \omega_{IOy}^2\right) \qquad (\text{ft/sec})^2$$

A functional flow diagram for the latter mechanization is shown in Figure 1-8.

Navigation computations are performed in double precision, fixed point arithmetic (component compensation is single precision). Attitude per se is not extracted in the unaided navigation mode.

1.6 CAROUSEL-IV Inertial Navigation Computations

1.6.1 General

The Carousel-IV inertial navigation system INS is manufactured by the Delco Electronics Division of General Motors, for navigation and steering of long range, commercial aircraft. The INS is comprised of the following units:

The Navigation Unit containing the inertial platform and associated electronics, the digital computer, the Control/Display Unit, the Mode Selector Unit and the Backup Battery Unit.

The computer provides navigation and steering information during flight (plus calibration and alignment during preflight). A variety of digital and analog inputs and outputs are available depending on the application. The computer is a serial, binary machine with 6144-word, 13-bit core memory. Instruction words are 12 bits plus parity bit, while data words are 24 bits including sign. Memory access time is 2.6 microseconds.

The platform is a four gimbal unit with the outer roll gimbal driven by the inner roll resolver so as to maintain the inner roll and pitch gimbals orthogonal. The stable element contains three single degree-of-freedom, floated, gas-bearing, rate-integrating gyros (AC651G), and three, single axis, linear, force rebalanced accelerometers. The stable element is subdivided into upper and lower halves: the upper half (the z platform) contains the azimuth gyro, the vertical accelerometer, the

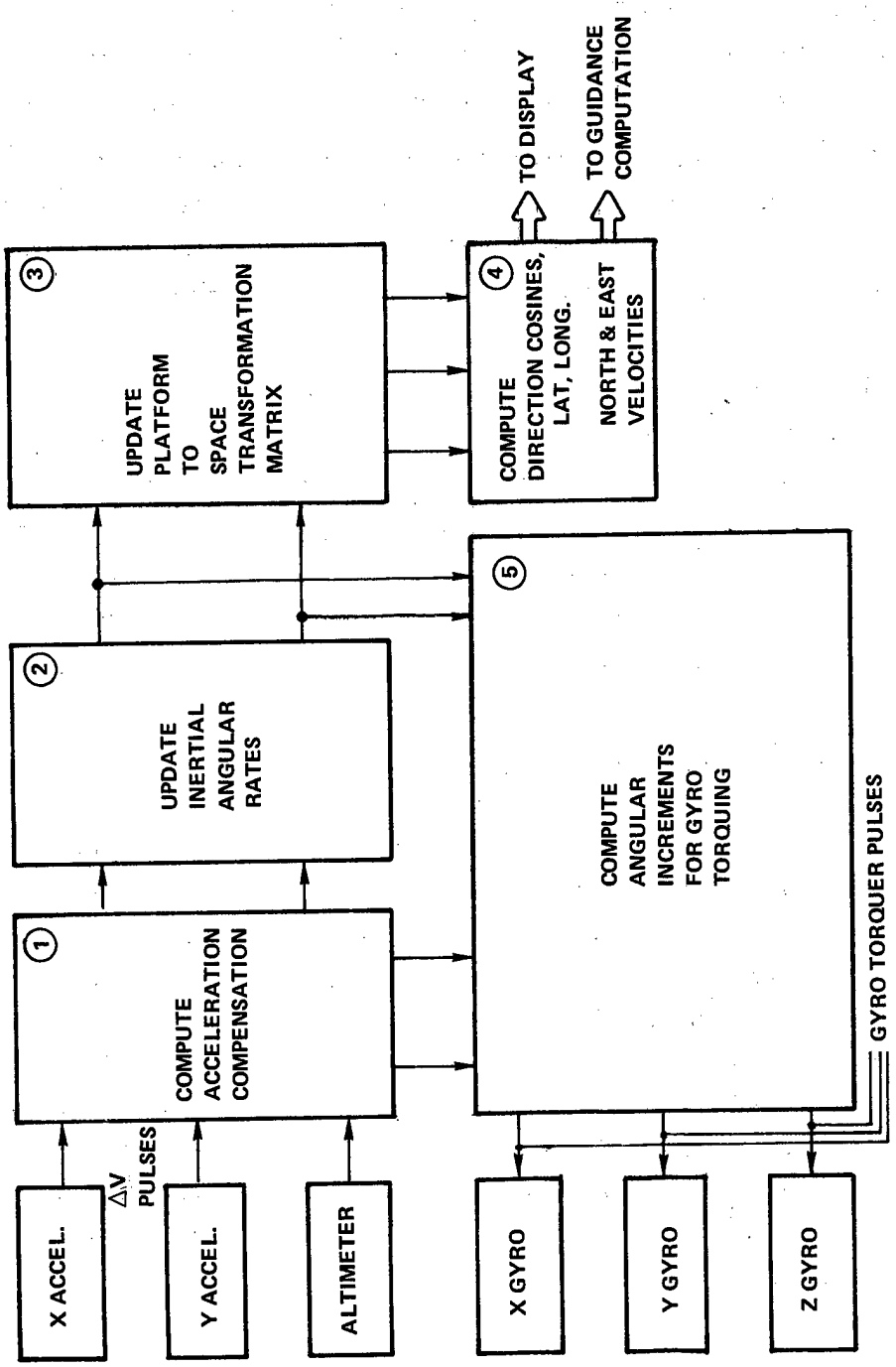


Figure 1-8. Functional Flow Diagram for Free-Azimuth Navigational Program

azimuth gimbal synchro and torque motor and the synchronous motor which drives the lower half (the x-y platform) at a constant rate (1 rpm) relative to the z platform; the lower half contains the two level gyros and accelerometers and two gimbal resolvers.

The vertical accelerometer output is analog (with subsequent A/D conversion) while the level accelerometer outputs are digital. The azimuth gyro is untorqued (free-azimuth) while the level gyros are pulse torqued via 20 Hz pulse-width modulated loops.

A second order baro-inertial vertical channel is employed with an added digital first order, low-pass filter applied to the barometric altimeter input.

This section describes the navigation computations for pure (baro) inertial mode of operation, based on documentation[8] [9] generated by Delco Electronics Division of General Motors (then at Milwaukee, Wisconsin).

1.6.2 Coordinate Frames

Four coordinate frames are used in the mechanization of position, velocity and attitude computations. The inertial instrument axes (platform axes) are designated x, y and z. The leveling process aligns z along the local gravity vector, and x and y rotate about the z axis at a rate ω_z with respect to inertial space (nominally 1 rpm). The u, v, z frame is a free-azimuth local-vertical frame used as a computational frame during the navigate mode. The MNO frame is an earth-centered, earth-fixed, right-handed frame with N along the earth's spin axis and M and O in the equatorial plane with the O axis passing through the Greenwich Meridian. These coordinates can most easily be visualized by reference to Figure 1-9.

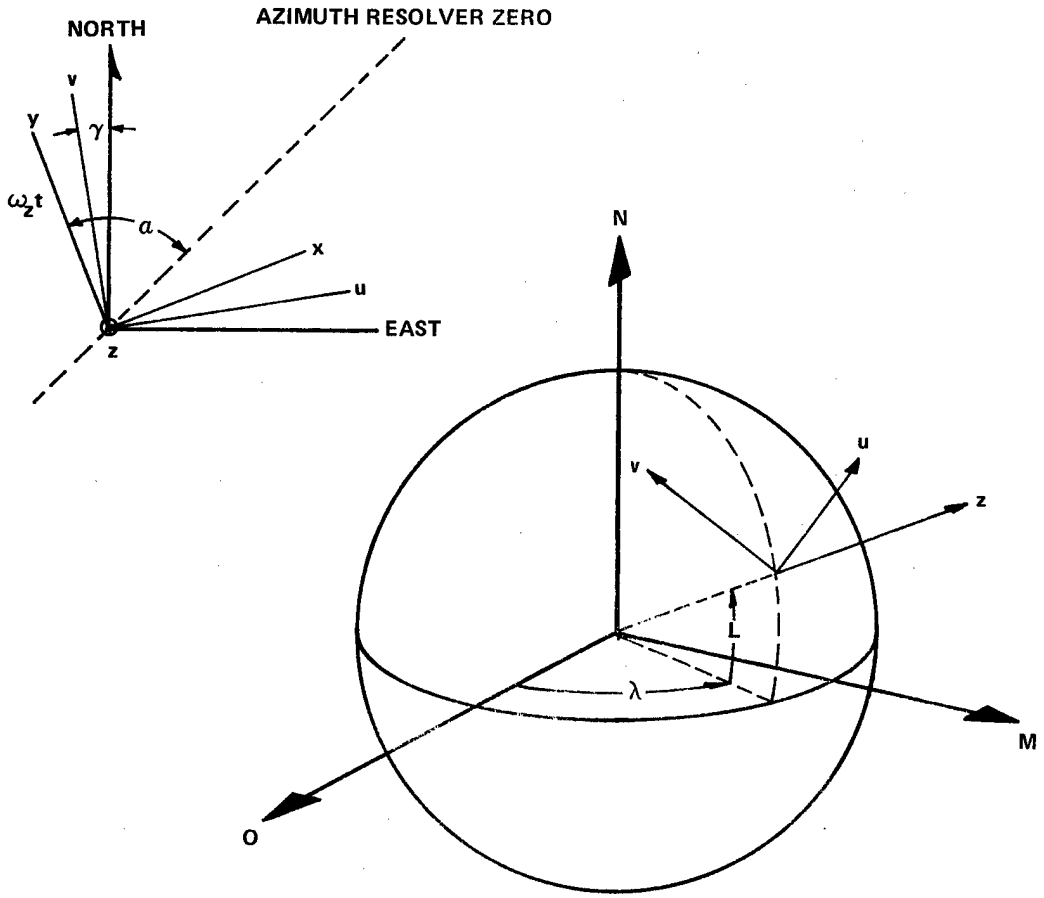


Figure 1-9. Carousel IV Coordinate Frames

1.6.3 Position and Velocity Computations

The Carousel IV position and velocity computations are performed in a free-azimuth, geodetic vertical computational frame. The computations are performed at three iteration rates:

1. The minor or interrupt loop performs incremental velocity resolutions and gyro torquing. The minor loop iteration rate is 50 milliseconds.

2. The intermediate or middle loop performs all horizontal velocity, ellipticity, and direction cosine computations in addition to two autopilot signals. The middle iteration rate is 200 milliseconds.

3. The remainder of the computation is done in the major loop at an iteration rate of 600 milliseconds.

At the minor loop frequency, 20 Hz, the level accelerometer ΔV's are transformed from the rotating x-y platform frame to the free-azimuth frame where they are summed for four cycles. The level gyro torquing rate is also updated. Every 200 milliseconds (5 Hz) the horizontal velocity computations are performed and the incremental velocity and the computed velocity increments are added to the old velocity. The vehicle rates are then computed and the direction cosine matrix is updated. Every 600 milliseconds (1 2/3 Hz) the position and the "wander" angle are extracted from the direction cosine matrix, the vertical channel computations are performed, the free-azimuth level velocities are transformed to the local-vertical north frame and the other computations required for control and display are performed.

The gravity computations employ the gravity model represented by the following equation:

$$g_z = 32.0881\ (1 + 5.2648 \times 10^{-3} \sin^2 L - 0.9558 \times 10^{-7} h)\ \text{ft/sec}^2$$

where L is the geodetic latitude and h is the altitude.

"Attitude" computation of heading consists of removing the wander angle from the encoded gimbal azimuth resolver reading. Pitch and roll are always available as analog gimbal resolvers outputs and are never digitized.

In summary, the "high" frequency (minor loop) computations are required only because the level (x-y) platform is rotating at 1 rpm. The x-y platform is rotated at this rate to average out many of the level gyro and accelerometer errors. The minor loop computations could be performed at the intermediate loop rate if the free-azimuth platform were not "carouseled".

1.7 SPN/GEANS Inertial Navigation Computations

1.7.1 General

The Standard Precision Navigator/Gimbaled Electrically Suspended Gyro Airborne Navigation System (SPN/GEANS) is a direct descendant of the GEANS (AN/ASN-101) high precision, aircraft inertial navigation system, both designed and manufactured by Honeywell. GEANS consists of the Inertial Measurement Unit (IMU), the Interface Electronics Unit (IEU), the Digital Computer Unit (DCU - a Honeywell HDC-601), and the Control and Display Unit (CDU). For SPN/GEANS, the latter two items, i.e., the DCU and the appropriate operator interface are government furnished equipment (GFE). The GEANS CDU HDC-601, is functionally an airborne equivalent of the Honeywell DDP-516 with an 8K memory (see SIRU description in section 1.8). For SPN/GEANS, the GFE DCU is manufactured by ROLM. In either case the prime interface between the IMU and IEU, and the CDU and other avionics subsystems is via a serial data bus (SDB).

The IMU employs a four gimbal, all attitude space-stabilized platform. It employs two, two-degree-of-freedom, electrically suspended gyros (ESG) with the redundant axis of one ESG caged (by employment of redundant axis torquing, RAT). Three single axis, hinged pendulum, flexipivot accelerometers (GG177) measure specific force. Pulse-width torquing is applied to the RAT axis; accelerometer rebalance is via "bang-bang", hundred percent duty-cycle loops.

Vertical channel (radial) damping computation employs a constant gain, second-order loop. Doppler radar measurements (when available) are applied via a Kalman filter for velocity damping and/or in air alignment. Position fixes are applied by the same means.

This section describes the navigation computations of position, velocity, attitude and attitude rate in the pure baro-inertial mode of operation. The description is based on the SPN/GEANS Computer Program Development Specification [2] generated by Honeywell Aerospace, St. Petersburg, Florida.

1.7.2 Coordinate Frames

The four primary coordinate frames employed for the navigation and attitude computations are the inertial reference frame (which is the computational frame), the local (geodetic) vertical north frame, the body (or vehicle) frame and the platform space-stabilized frame.

The computational frame (axes x_0, y_0, z_0) and the local vertical geodetic frame (axes x_2, y_2, z_2) are shown in Figure 1-10. An intermediate frame (axes x_1, y_1, z_1) coincide with the computational frame axes at the start of the inertial mode of operation, $t = 0$. As shown in Figure 1-10, L is the INS geodetic latitude, $\Delta\lambda$ is the INS longitude change during the flight and ω_e is the magnitude of the earth's angular velocity with respect to the inertial space.

The vehicle body frame (axes x_5, y_5, z_5) is shown in Figure 1-11. The vehicle body axes coincide with the local geodetic frame axes for a leveled vehicle with the fuselage pointing north.

The platform reference axes have a fixed orientation with respect to the computational frame axes.

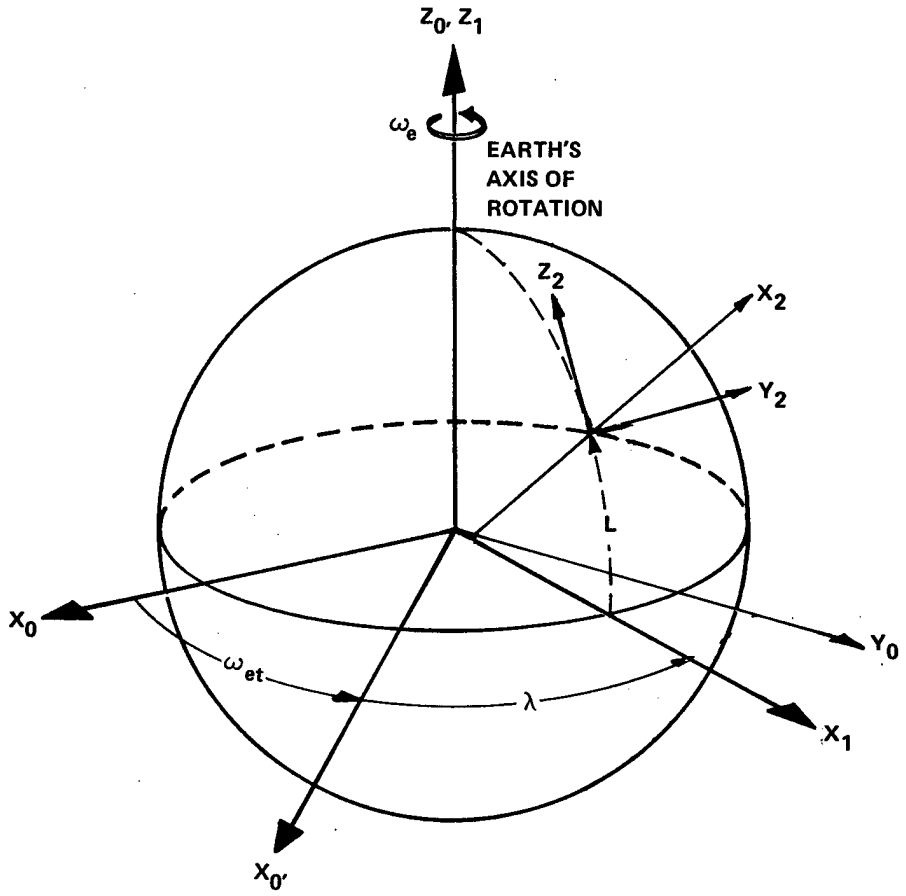


Figure 1-10. Frames Used in SPN/GEANS Computations

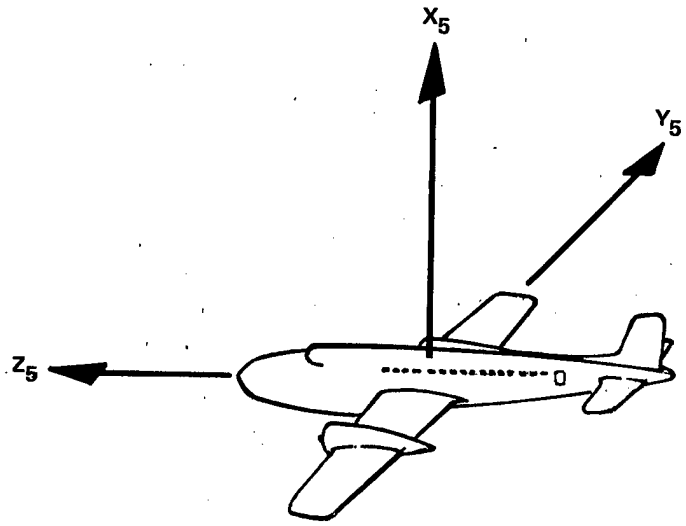


Figure 1-11. Vehicle Frame Axes Used In SPN/GEANS Computations

1.7.3 Position and Velocity Computations

The SPN/GEANS position and velocity computations are performed in the space-stabilized computational frame, as shown in Figure 1-12. The velocity increments, $\Delta\bar{V}_0$, sensed by the accelerometers and resolved along the computational frame axes are combined with the gravity contributions, $\bar{G}\Delta t$, and with the barometric damping term, $k_1\Delta h$. The computed inertial velocity, $\bar{V}_0$, is used in computation of the local geodetic velocity components, $\bar{V}_2$. The $\bar{V}_0$ is also used with the barometric damping term, $K_2\Delta h$, in the computation of the inertial position, $\bar{R}_0$. The $\bar{R}_0$ is used in the gravity computations, in the geodetic velocity computations and in the geodetic position computations (latitude, L, longitude, λ, and altitude h).

Position and velocity computations are performed every 125 milliseconds (8 Hz).

1.7.4 Attitude Computations

The SPN/GEANS attitude computations are shown in a functional block diagram, Figure 1-13. THe attitude angles are computed by solving the following matrix equation:

Attitude angles matrix = Transformation matrix from the vehicle body axes to the geodetic axes

The attitude angles matrix is a function of the vehicle yaw, pitch and roll angles.

The transformation matrix from the vehicle body axes to the geodetic axes consists of the transformation from the vehicle body axes to the platform axes followed by the transformation from the platform axes to the geodetic axes. The former transformation is determined from the gimbal angles; the latter from the computed position and time of the inertial mode of operation.

The attitude computations are performed every 31.25 milliseconds (32 Hz).

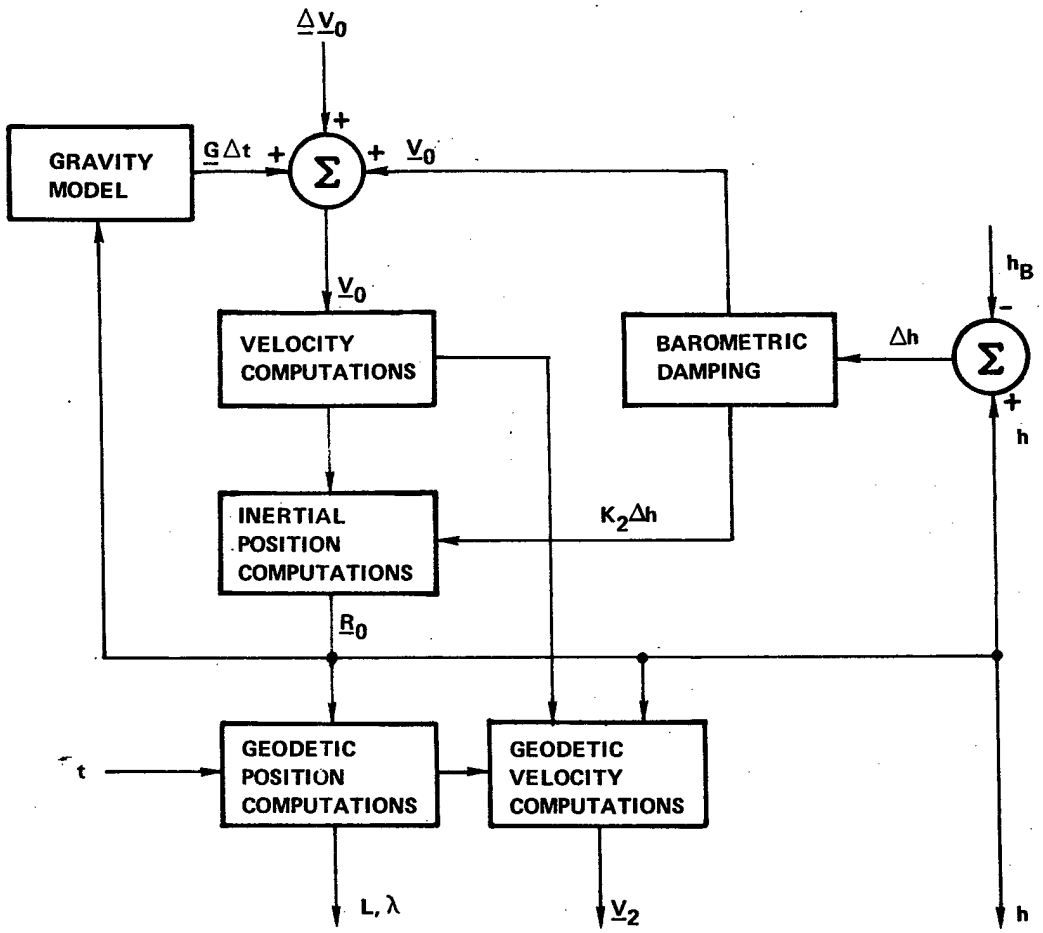


Figure 1-12. SPN/GEANS Position and Velocity Computations - Functional Block Diagram

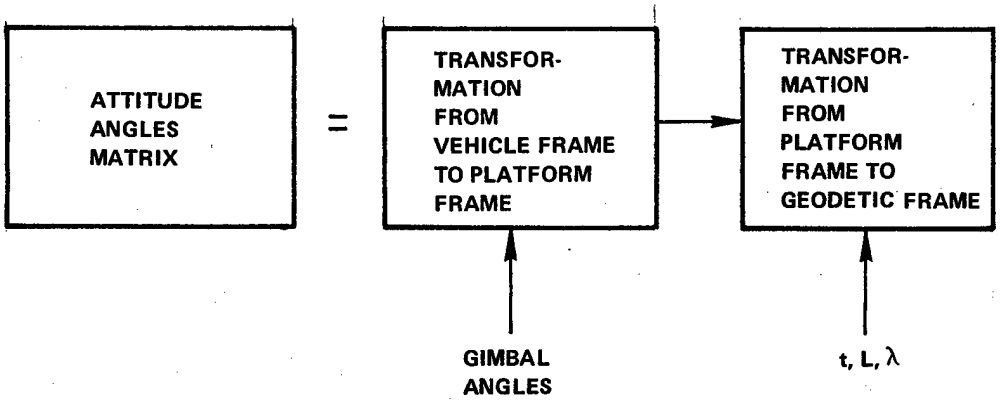


Figure 1-13. SPN/GEANS Attitude Computations - Functional Block Diagram

1.8 H-429 (SIGN III) Inertial Navigation Computations

1.8.1 General

H-429 is a designation for a member of a family of Honeywell designed and built strapdown inertial navigation systems, which was used in the SIGN III Kalman Inertial Doppler System flight tests at Holloman Air Force Base during 1970 - 1971 period.

The INS uses the SIGN III computer, manufactured by Honeywell to perform the attitude, navigation and Doppler/Inertial/Checkpoint Kalman filter computations for the system as well as driving the recorders and printers for in-flight and post-flight evaluation. The primary external input required for the navigation function is barometric altitude. The computer is a 4096-word, 20-bit, fixed-point, twos complement, machine with a 2-microsecond cycle time.

The IMU consists of a strapdown block containing three, one-degree-of-freedom, gas bearing gyros (GG334) and three single axis, flexipivot, hinged pendulum accelerometers. The inertial components, all of Honeywell manufacture, are pulse-rebalanced at 3.6 kHz; they have a high angular rate and high g capability compatible with high performance aircraft applications.

The input axes of the inertial sensors nominally coincide with the aircraft body axes. Navigation computations are performed in a geocentric vertical north pointing frame, hence the system is not suitable for use in the immediate vicinity of the earth's poles. The barometeric altimeter is incorporated into a constant gain, second order, baro-inertial (geocentric) vertical channel. Doppler velocities and checkpoint fixes are incorporated into the system via the Kalman filter.

This section describing the navigation computations for the pure (baro-inertial) mode, is based on internal Honeywell documentation[10][11].

1.8.2 Coordinate Frames

This section describes the frames used in the navigation computations of position, velocity, and attitude.

The frames used in the above computations are the body (aircraft) or platform frame and the geocentric local vertical north pointing frame. The x-axis of the body frame is towards the belly, the y-axis is towards the nose and the z-axis is towards the right wing. The geocentric local vertical north pointing frame has the x-axis towards the center of the earth, the y-axis is north in the geocentric level frame, and the z-axis is east. This is the computational frame.

The transformation from the body frame to the computational frame represented by the attitude matrix, C_b^c , is given by:

$$C_b^c = x\left(-\psi_{GC}\right) z\left(-\theta_{GC}\right)\ y\left(-\phi_{GC}\right)$$

The attitude angles, ψ_{GC}, θ_{GC}, ϕ_{GC}, defined with respect to geocentric frame, differ from those defined relative to the geodetic vertical by up to 3.3 milliradians at 45° latitude.

1.8.3 Position and Velocity Computations

- The H-429 inertial navigation computations are performed in the geocentric local vertical, north-pointing frame.
- The basic navigation and attitude matrix update computational cycle is 20 milliseconds (50 Hz).

The attitude matrix update employs a second order quaternion update algorithm using the "total $\Delta\theta$" method, described in Section 2, and conversion to the direction cosine matrix form.

The angular velocities computed are the inertial angular velocities of the vehicle in the computational frame (inertial velocities expressed in the computational frame).

In computation of the east component of the aircraft velocity with respect to the earth, the earth's inertial velocity is subtracted from the computed total inertial angular velocity. The aircraft north and vertical components of velocities with respect to the earth are computed in the geocentric frame with no conversion to the geodetic frame.

Latitude and longitude are obtained by integrating the angular rates. Latitude is converted to the geodetic frame for display. The "gravity" model used contains the geocentric north and vertical components of the mass attraction. The equations are (10)

$$G_D = \frac{-\mu}{R^2}\left\{1 + \frac{3J_2R_{EQ}^2}{2R^2}\quad (1 - 3\sin^2\lambda)\right\}$$

$$G_N = \frac{\mu\ 3J_2\ \sin\lambda\ \cos\lambda}{R^4}$$

where the constants, μ and J_2, are

$$\mu = 1.407654 \times 10^{16}\ ft^3/sec^2$$

$$J_2 = 1.0828 \times 10^{-3}$$

and

$$R_{EQ} = 2.0925738 \times 10^7\ ft$$

R is the computed present geocentric radius to vehicle (ft) and λ is the computed present geocentric latitude.

R is initialized from

$$R_o = R_{EQ}\left\{1 - \varepsilon\sin^2\lambda\left(1 + \frac{3}{2}\varepsilon\cos^2\lambda_o\right) + H_o\right\}$$

where o's as subscripts denote initial values and $\varepsilon = 1/298.3$, and H_o is the initial barometric altitude. Thereafter R is computed from

$$R = R_o - \int_o^t V_D(\tau)\,d\tau$$

The present altitude, H, is computed from a similar equation

$$H = H_o - \int_o^t V_D(\tau)\,d\tau$$

where $V_D(\tau)$ is the computed geocentric vertical velocity at time τ. Note that in equations of $R - R_o = H - H_o$, the variation in R as a function of latitude for a constant barometric altitude, H_B, is not taken into account. As a result an altitude reference error of the order of εR_{EQ} $(\sin^2\lambda_o - \sin^2\lambda)$ will occur at the input to the second-order vertical damping loop.

1.9 SIRU Inertial Navigation Computations

1.9.1 General

SIRU is the acronym for the Strapdown Inertial Reference Unit, designed and built by CSDL under contract to NASA/MSC. It is a redundant modular instrument package developed to achieve and investigate high reliability guidance and navigation concepts.

For fixed based tests, the SIRU is interfaced with a Honeywell DDP-516 digital computer with a 16K (16,384-word) core memory plus two discs. For the current flight test program, SIRU uses a dual, ruggedized, Honeywell DDP-316 installation, each with a 12K (12,288-word) core memory. Both computers are 16-bit, fixed point, twos complement machines; the prime operational difference is the memory cycle time of 0.96 microsecond for the DDP-516 and 1.72 microseconds for the DDP-316 computers.

The instrument package contains six single-degree-of-freedom, floated, gas bearing, rate-integrating gyros (size 18, model B, IRIG) operating in a pulse-torqued mode, and six single axis, floated-pendulous accelerometers (16 PM, PIPA) operating in a 9.6 kHz pulse-on-demand, pulse-rebalance mode. The input axes are each normal to one face of a regular dodecahedron, but are packaged in prealigned modules permitting simple installation in a rectangular shaped frame.

From the redundant sensors an equivalent orthogonal triad is defined and used in navigation computations. A second order, constant gain, baro-inertial vertical channel is employed with the altitude set to 10,000 feet.

This section describes the navigation computations of position and velocity in the pure-inertial mode. Attitude angles are not extracted. The description is based on reports generated at CSDL [12][13][14].

1.9.2 Coordinate Frames

The coordinate frames used in position and velocity computations in the navigation mode are defined in this section.

The equivalent orthogonal sensor triad is nominally parallel to the vehicle or body frame. The x-axis is parallel to the vehicle yaw axis - positive up. The y-axis is parallel to the vehicle pitch axis - positive towards the right wing. The z-axis is parallel to the vehicle roll axis - positive forward.

The "local-level" computational frame employed for the navigation computations is geodetic vertical, north pointing. The body and computational frames coincide for a vehicle geodetically leveled heading north. The x-axis of the computational frame is directed along the geodetic, vertical - positive "up". The y-axis is tangent to the meridian positive north.

It should be noted that, while the two frames defined above are the same as the frames employed in the H-429 (SIGN III) navigation computations, described in section 1.8, the axes are rearranged, as inevitably seems to be the case.

The body to computational frame transformation C_b^c in this case is

$$C_b^c = x(\psi)\ y(-\theta)\ \ z(-\varphi)$$

where the geodetic attitude angles are, ψ, θ, φ .

1.9.3 Position and Velocity Computations

The SIRU inertial navigation computations are perfomred in a local geodetic vertical, north-pointed frame. This renders the system unsuitable for use in the immediate vicinity of the earth's poles.

In SIRU navigation computation the radii of curvature, meridional and prime vertical, r_m and r_p are given by:

$$r_m = r_o\ (\ 1 - 2e\ \cos^2 L)$$

and

$$r_p = r_o\ (1 + 2e\ \sin^2 L)$$

where

$$r_o = r_e\ (1 - e\ \sin^2 L)$$

and r_o is the sea level radius from the center of the earth to the present position.

r_e is the equatorial sea level radius (6378163 meters)

e is the ellipticity of the meridional ellipse (1/297)

L is the geodetic latitude

Latitude and longitude rate, $\dot{L}$ and $\dot{l}$, are given by

$$\dot{L} = -V_N/(r_m + h)$$

$$\dot{l} = V_E/(r_p + h)\ \cos L$$

where V_E and V_N are the east and north components of the aircraft velocity with respect to the earth. The attitude matrix is updated every 20 milliseconds (50 Hz) using the "total $\Delta\theta$" method and third order quaternion update algorithm and normalization similar to that described in section 2. After conversion to the direction cosine matrix form, the updated matrix is used to transform the ΔV's accumulated over a 20 millisecond period to the local vertical frame to update the V. The ΔV and $\Delta\theta$ accumulation times are staggered so that the matrix values are appropriate to the midpoint of ΔV interval.

The position and velocity computations are performed at a 1 Hz rate and are earth relative (rather than inertial relative as was the case for SIGN III) and h is an externally supplied altitude (set to 10,000 feet for flight test). Since the vertical channel does not supply altitude but only altitude rate, $\dot{h}$, to the inertial computations, the gravity model is of lesser importance and only a constant vertical component of gravity is used.

No attitude computations are performed in the SIRU mechanization.

SECTION 2

DETAILED NAVIGATION EQUATION STRUCTURE

2.1 Introduction

This section presents the equations which are mechanized in the navigation computer portion of the numerical simulator for any of a local vertical wander azimuth (LVWA), a space stable (SS), or a strapdown (SD) IMU. While computational flow diagrams are included for both inertial (SS) and LVWA computational frames, only the LVWA navigation algorithm is developed in detail since this is the "selected" computational frame (see Section 5 of this volume). Another area requiring some exposition is the strapdown computations, which, again, are developed in detail only for use with a LVWA computational frame. Similarly, the attitude predictor filter is described in some detail. The remainder, consisting mainly of transformations, are referred to and relegated to the Appendices.

2.2 Navigation and Attitude Computations

2.2.1 General

The navigation and attitude computations for all three INS types and either computational frame naturally subdivide into three gross modules:

(1) Transformation (T)

(2) Navigation (N)

(3) Attitude (A)

2.2.2 Tranformation and ΔV Accumulation

The function of the first module, T, is the generation of the orthogonal transformation from the platform (P) frame to the computational (C) frame and the subsequent transformation of the incremental velocities accumulated over one sampling interval, Δt_T, from the platform frame to the computational frame. Reduced to simplest terms, it mechanizes the "equations".

(a) Generate C_p^c

(b) $\underline{\Delta v}^c = C_p^c \, \underline{\Delta v}^p$

Depending on the nature of C_p^c, the platform frame incremental velocities, $\underline{\Delta v}^p$, may be transformed from the platform frame to the computational frame more often than once per navigation computational cycle, Δt_N, in which case the ratio, $\Delta t_T / \Delta t_N = 2^m$, where m = 1, 2, 3, ... An additional function is defined in this case, i.e., the summation of the incremental velocities in the computational frame.

$$(c) \quad \underline{\Delta v}^c_{(N)} = \sum_{i=1}^{i=2^m} \underline{\Delta v}^c_i$$

For a LVWA INS using a LVWA computational frame, the platform (p) frame and the computational (c) frame are identical, i.e., $C_p^c = I$, and neither of steps (a) or (b) is required. If the sampling interval is less than the navigation computational cycle it would be necessary to sum $\underline{\Delta v}$'s as indicated in step (c).

For a SD INS using a LVWA computational frame, the platform (p) frame is the body (b) frame. The transformation required is C_b^c, which is generated as indicated in Section 2.3.3, the section on strapdown computations.

For a SS INS using a LVWA computational frame, the platform frame is related to the computational frame by a transformation which is expressed as the product of three rotation matrices, viz:

$$C_p^c = C_e^c \, C_i^e \, C_p^i$$

In the present study, the platform to inertial frame transformation, C_p^i, is a constant since the platform is considered to be non-drifting. C_p^i is supplied by the initial conditions.

The transformation from the inertial (i) to the earth fixed (e) frame, C_i^e, is a single axis rotation, Z ($\omega_{ie}t$). It accounts for the rotation of the earth from the start of navigation until the start of the present navigation computation cycle. Its initial value is I. The transformation from the earth fixed (e) frame to the computational (c) frame is the transpose of the "direction cosine" matrix computed in the LVWA navigation algorithm and is appropriate for the start of the present navigation cycle, but uses the "gyro torquing" signals to extrapolate C_e^c to the middle of the navigation computational cycle.

To minimize the computations required, the incremental velocities in the SS platform frame are premultiplied successively by each of the three individual rotation matrices, C_p^i, C_i^e, C_e^c to obtain $\underline{\Delta v}^c$. This operation requires three matrix-by-vector multiplications or 27 multiplies and 18 adds, whereas forming C_p^c from three matrices and then transformating $\underline{\Delta v}^p$ requires 63 multiplies and 42 adds.

The sequence of matrix and vector operations is indicated in the computational flow diagrams for the three INS types with a LVWA computational frame (Figures 2-1, 2-2, 2-3a, and 2-3b).

The corresponding transformation operations for the three INS types with a SS computational frame are indicated in the computational flow diagrams (Figures 2-4, 2-5 and 2-6). In increasing order of complexity of transformation, the systems rank SS, SD and LVWA.

2.2.3 Navigation

The function of the second, N, module is to generate the geodetic, local vertical, north-pointing, velocity and position components from the incremental velocity, $\underline{\Delta v}^c$, and the initial conditions. A non-inertial altitude indication, generally barometric altitude,is also required. These operations are embodied in the Navigation Algorithm with Vertical Damping and the Output Conversion NAVD/OC. The routines included in the NAVD/OC are tabulated in Tables 2-1 and 2-2.

With a LVWA computational frame, the outputs not only include the LVN position and velocity but also the wander angle, α, and the gyro torquing signals, $\underline{\omega}_{ic}^c$. All indicated computations are performed for all three INS types. The individual routines are outlined in Table 2-1.

With a SS computational frame, the outputs are LVN position, velocity, and inertial-to-LVN transformation, C_i^ℓ (or C_0^2). The individual routines are outlined in Table 2-4. Wander angle and gyro torquing signals are required only for a LVWA INS. Special computations required for the above system also contribute to the generation of the transformation from the (LVWA) platform frame to the computational (inertial frame) as required for the first (T) module.

2.2.4 Attitude

The function of the third, A, module is to generate the attitude (roll, pitch, heading) angles of the vehicle.

With the LVWA computational frame the readily available transformation is that from the body frame to the computational frame, C_b^c. As indicated in Appendix 4.4.7 this is not quite the attitude matrix, C_b^ℓ, but differs from it only to the extent that the heading "extracted" from this matrix is $(\Psi-\alpha)$. This is shown in the computational flow diagrams Figures 2-1, 2-2, and 2-3b.

With the SS computational frame, the body-to-LVN transformation, C_b^ℓ, is generated directly from gimbal angles (as was the case with a LVWA computational frame) and the "attitude-like" angles are extracted as before; "heading" is corrected by the addition of the wander angle--see Figure 2-5.

Attitude will be computed at the same frequency as the navigation computations or at some binary multiple of the navigation frequency. In the latter case, it may be necessary to extrapolate the inertial-to-LVN transformation or the wander angle. Regardless of the computational frame, the three steps in attitude computation and filtering are:

1. Form C_b^ℓ (or C_b^c)
2. Extract "attitude" angles (correct heading if necessary)
3. Filter and predict attitude

The attitude filter-predictor will be described in Section 2.5.5.

Table 2-1. Local Vertical Wander Azimuth Navigation Algorithm with Vertical Damping

1	Update platform frame velocities.
2	Add vertical damping terms.
3	Compute angular velocities about level axes.
4	Update direction cosine matrix (opt. 2nd order).
5	Orthonormalize direction cosine matrix (1st order).
6	Compute earth's rate signals (angles).
* 7	Compute gyro torquing signals.
8	Compute angular velocity [for $(\bar{\omega}x\bar{v})$].
9	Compute $(\bar{\omega}x\bar{v})\ \Delta t$ terms (for velocity).
10	Compute altitude and vertical damping.
11	Compute gravity components.
* 12	Extract position and wander angle.
* 13	Form wander angle matrix and LVN velocities.

*Shown as separate blocks in computational flow diagrams.

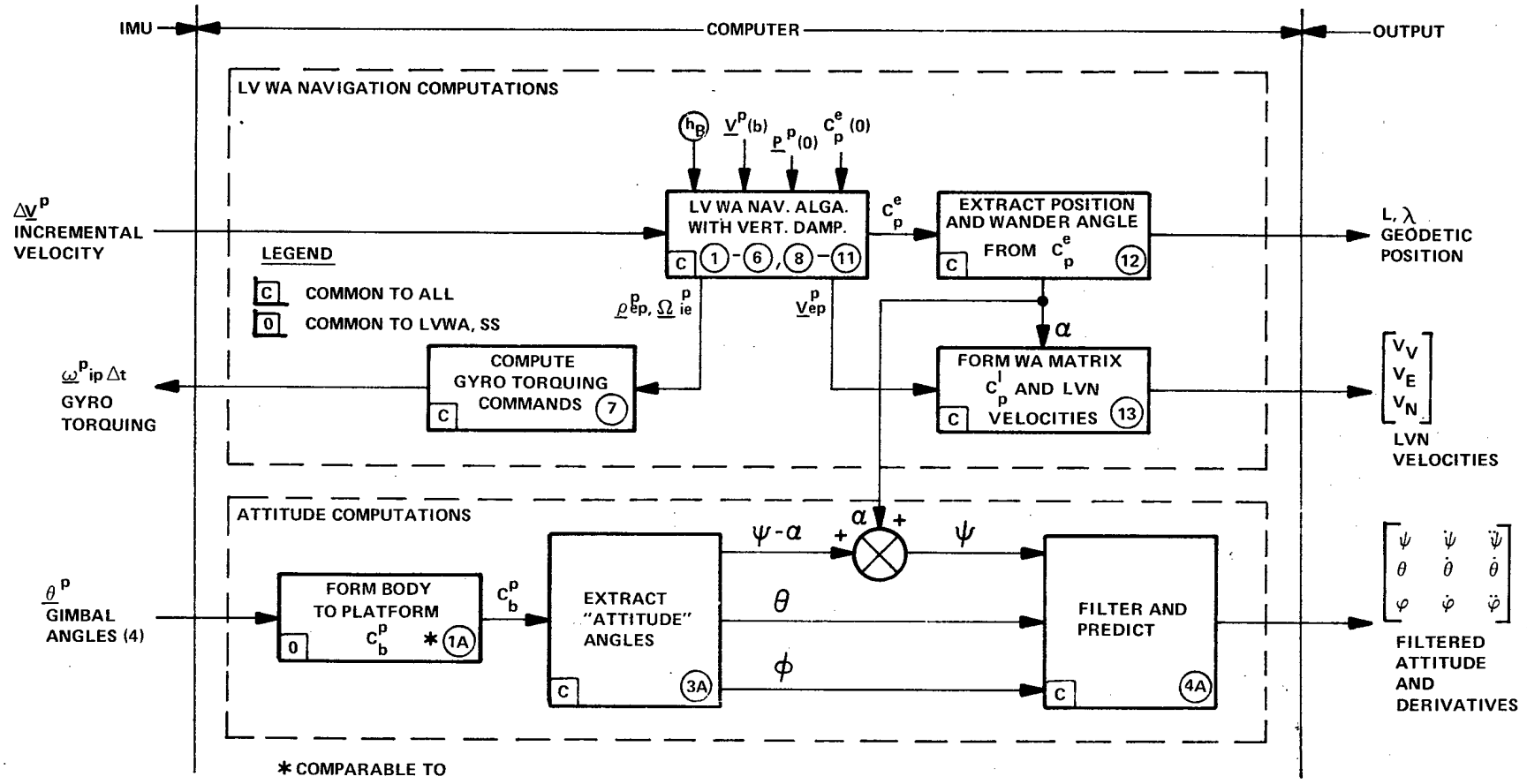


Figure 2-1. Local Vertical Wander-Azimuth INS Computational Flow Diagram with LVWA Computational Frame

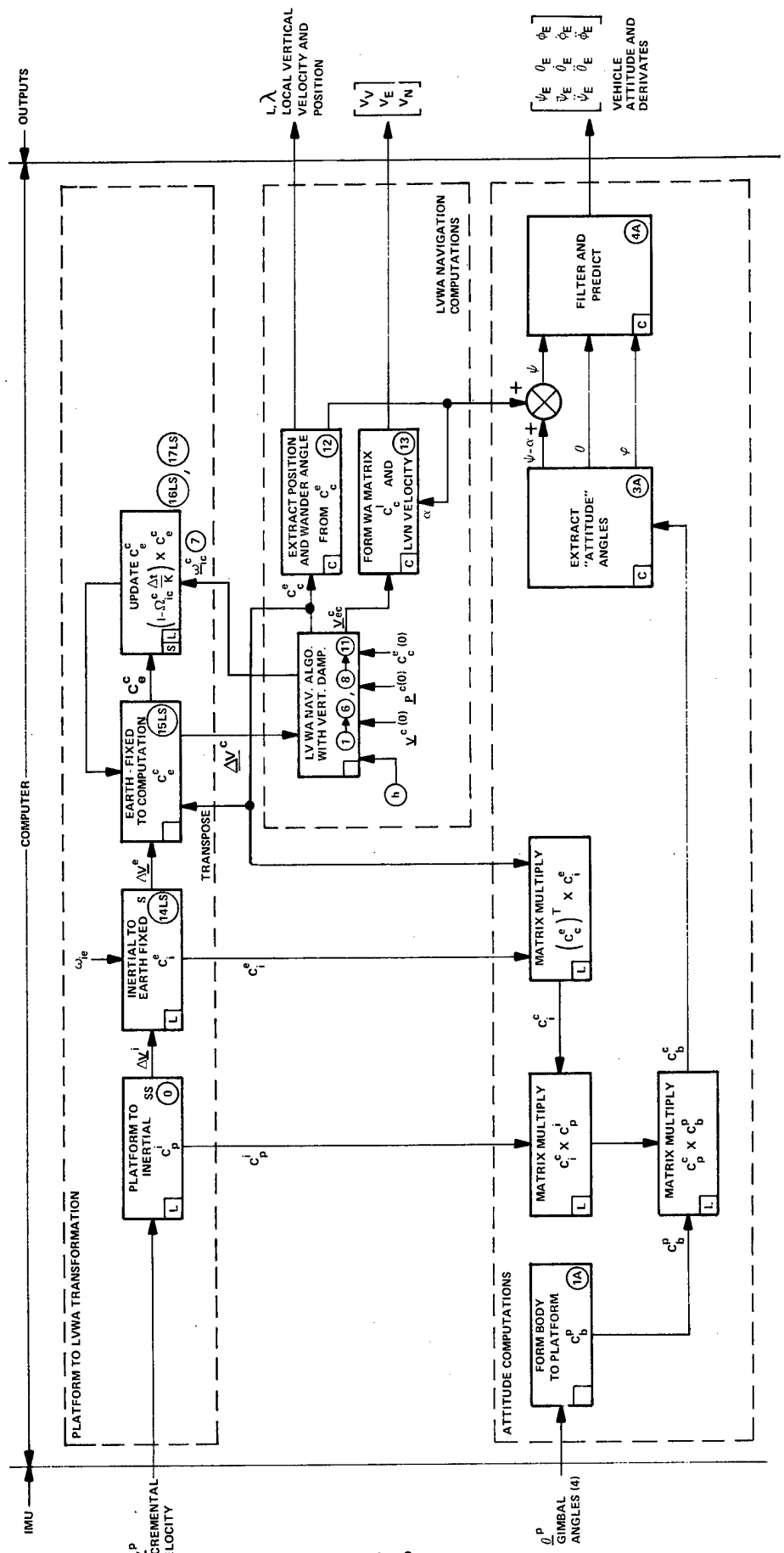
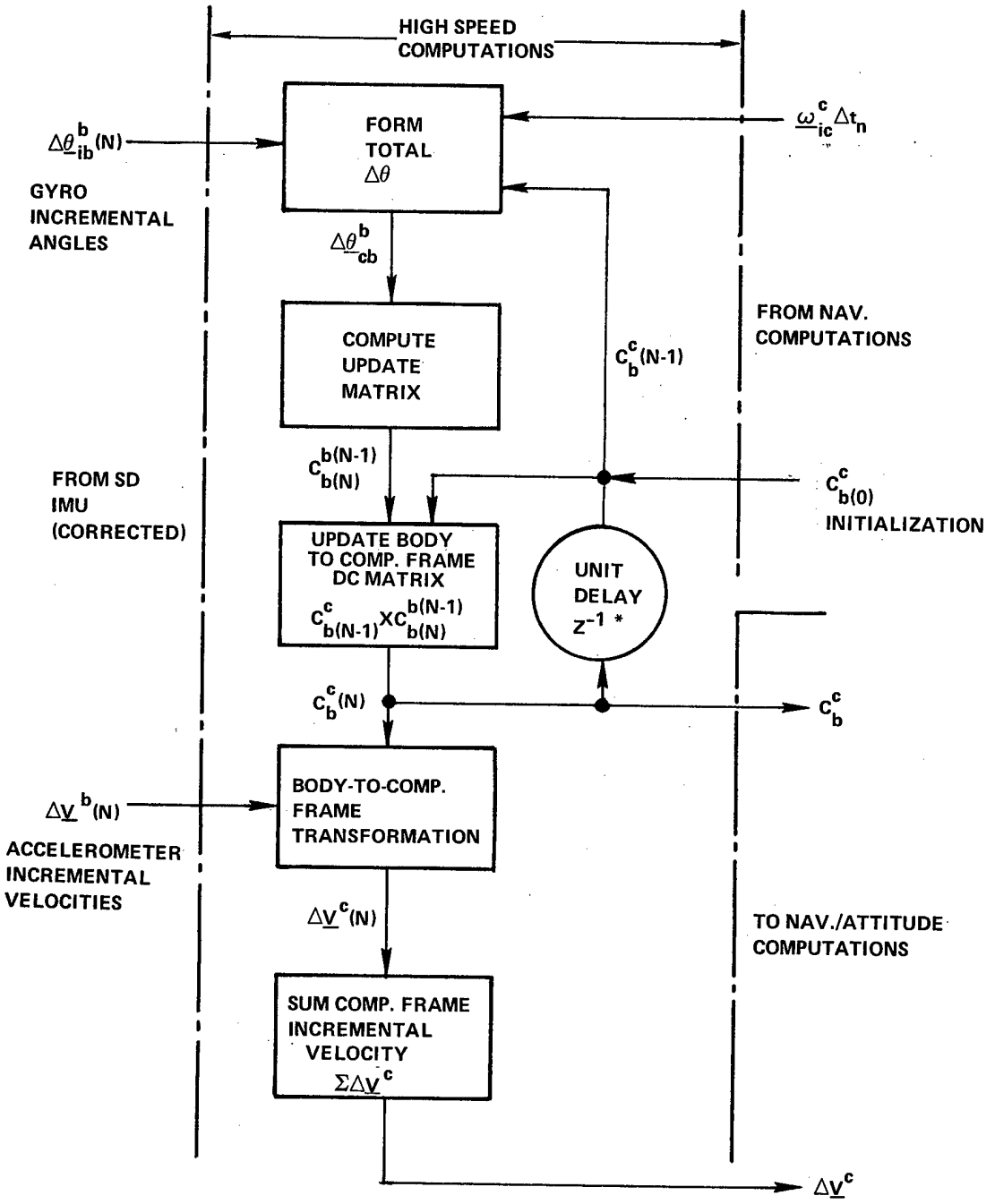


Figure 2-2. Space Stable INS Computational Flow Diagram with LVWA Computational Frame



*NO COMPUTATIONS – INSERTED TO EMPHASIZE FOR USE OF C^{c}_{b} FROM PREVIOUS HIGH SPEED COMP. CYCLE.

Figure 2-3a. High Speed Computational SD System-LVWA Navigation Frame

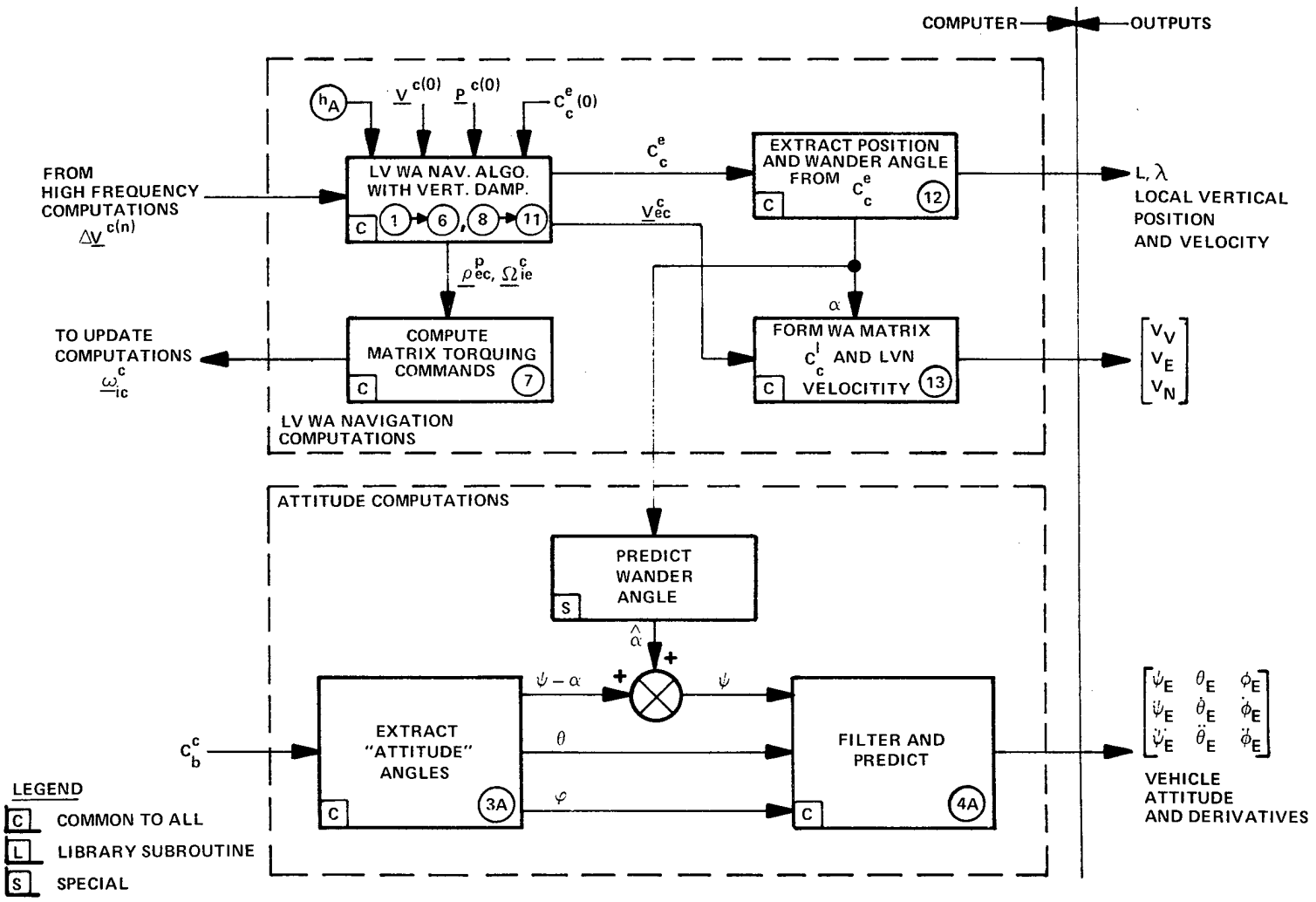


Figure 2-3b. Strapdown INS Computational Flow Diagram with LVWA Computational Frame. Navigation and Attitude (Low Speed) Computations

Table 2-2. Space Stable Navigation Algorithm with Vertical Damping — Subtracting East Velocity of Earth

1	Extrapolate position to mid comp. cycle.
2	Compute mass attraction
3	Compute vertical damping for position computation
4	Update velocity for position computation
5	Update position
6	Update velocity
7	Remove east velocity of earth.
8	Compute inertial and reference radii difference.
	Transformation to Local Vertical and Local Vertical (North) Outputs
9	Compute latitude and longitude
10	Form inertial to (geodetic) local vertical transformation
11	Transform inertial-referenced velocity to local vertical

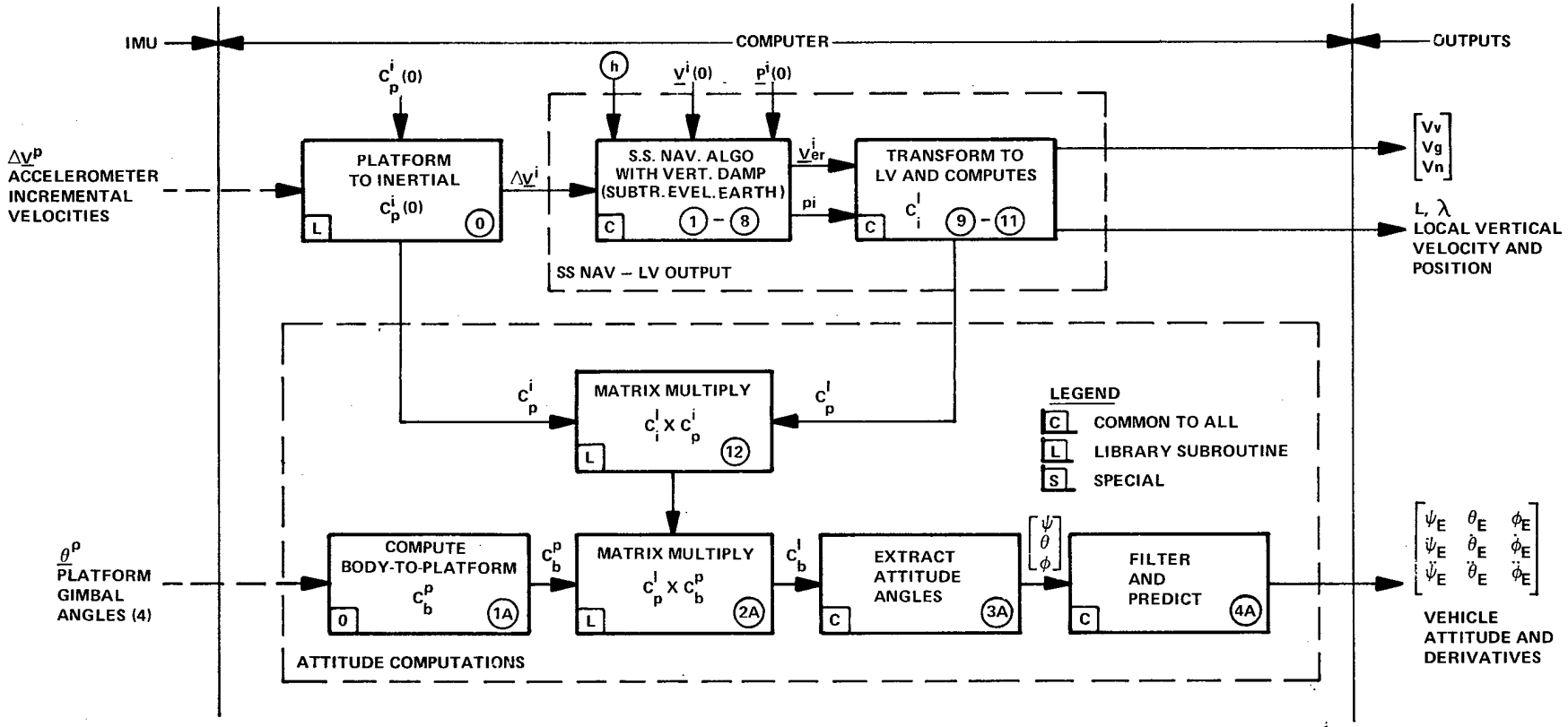


Figure 2-4. Space-Stable INS Computational Flow Diagram with SS Computational Frame

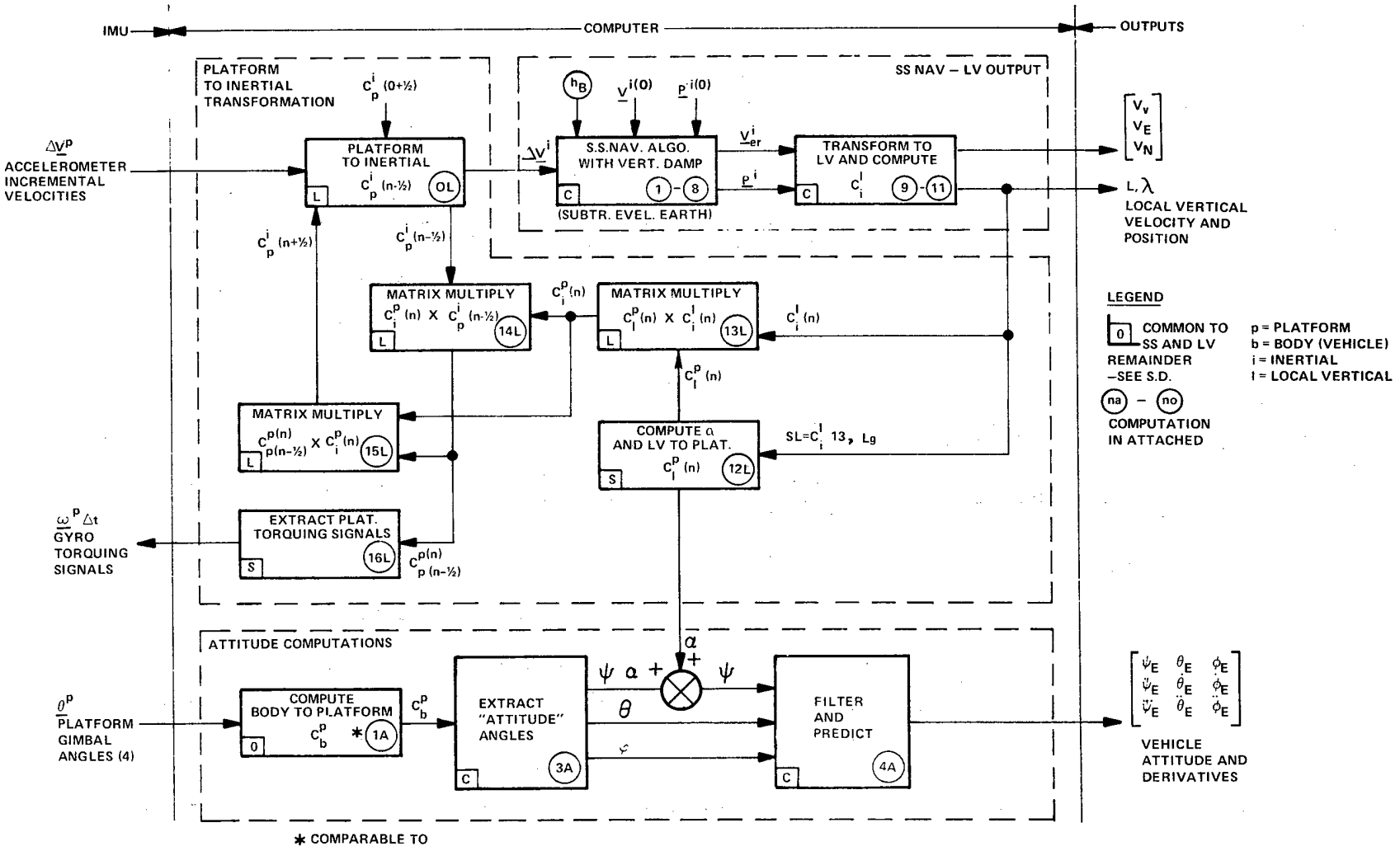


Figure 2-5. Local Vertical Wander-Azimuth INS Computational Flow Diagram with SS Computational Frame

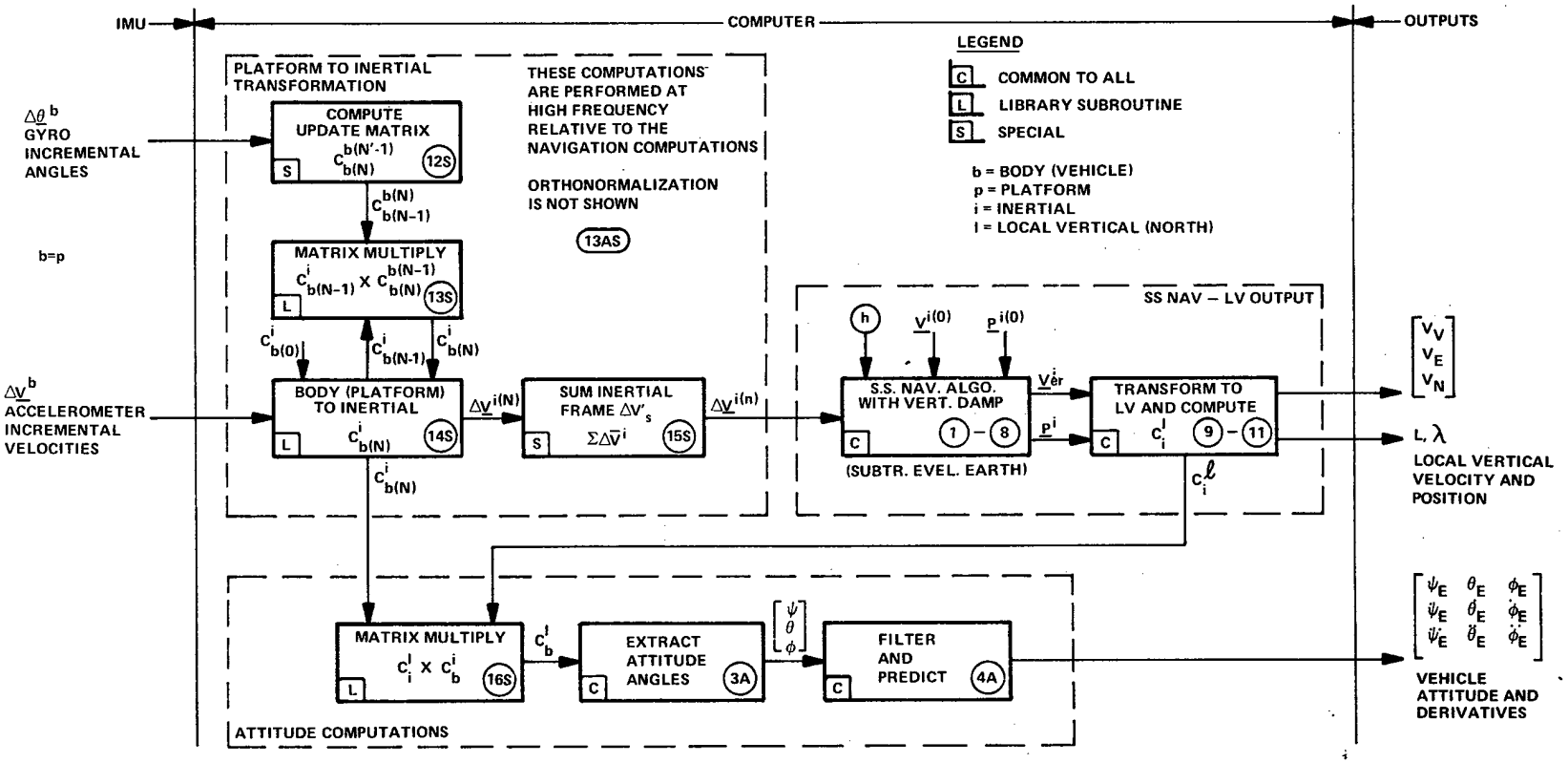


Figure 2-6 . Strapdown INS Computational Flow Diagram with SS Computational Frame

2.3 Transformation and Incremental Velocity Accumulation

Some further comments are required on the "phasing" of the incremental velocities, ΔV's and the transformation to the LVWA computation frame in the cases of the SS and SD IMU's.

2.3.1 Local Vertical Wander Azimuth IMU

Ideally, the platform of the LVWA IMU coincides with the LVWA computational frame. Hence, the ΔV's are merely summed over a navigation computation cycle and entered directly in the velocity update routine of the navigation algorithm.

In practice, the IMU and computational frames do not quite coincide with each other nor with true LVWA frame, due to gyro torquing errors. Thus, the Δv's from a LVWA IMU generally reflect the effects of infinitesimal rotation from their "true" values. The mechanization of the gyro torquing errors is a part of the LVWA IMU simulator portion of the numerical simulator.

2.3.2 Space Stabilized IMU

The comments on the transformation of the ΔV's from a SS IMU to the LVWA computational frame as described in Section 2.2.2 are directly applicable to the simpler, or "baseline" software shown in the computational flow diagrams.

However, when the "upgraded" navigation algorithm is used, an estimate of C_c^e (or $C_{2'}^{0'}$) at the mid-point of the next navigation computation cycle is generated, which obviates the necessity of "torquing" C_c^e (or its transpose) specifically for the purpose of transforming the ΔV's. The inertial to earth fixed transformation, C_i^e ($C_o^{o'}$) must then be computed at the mid-point of the

computation cycle rather than at the beginning as was formerly the case. The equations mechanized are indicated below:

$$\underline{\Delta v}_n^0 = C_{10}^0 \, \underline{\Delta v}^{10}$$

$$\underline{\Delta v}_n^{0'} = C^{0'}_{0_{n-1/2}} \, \underline{\Delta v}_n^0$$

$$\underline{\Delta v}_n^{2'} = (C^{0'}_{2'_{n-1/2}}) \, \underline{\Delta v}_n^{0'}$$

where $\underline{\Delta v}_n$, $\underline{\Delta v}_n^0$, $\underline{\Delta v}_n^{0'}$, $\underline{\Delta v}_n^{2'}$ are the incremental velocities accumulated over the nth computation cycle expressed in the SS platform, the inertial, earth fixed and LVWA frames respectively.

$C^{0'}_{2'_{n-1/2}}$ is the extrapolated transformation from the LVWA to the earth fixed frame, computed at the end of the $(n-1)^{st}$ computation cycle but extrapolated to the mid-point of the nth computational cycle.

$C^{0'}_{0_{n-1/2}} = Z\left[(\omega_{ie}(n-1/2)\ \Delta t_n)\right]$ is the transformation from the inertial to the earth fixed frame at the mid-point of the nth computational cycle

C_{10}^0 (or C_p^i) is SS platform-to-inertial transformation or SS alignment matrix.

In this particular case, fewer computations are required for the transformation of space stable platform ΔV's to the LVWA frame than for the basic "baseline" algorithm.

2.3.3 Strapdown IMU

The strapdown navigation equations may be mechanized either in the space-stabilized computational frame or in the local-level computational frame. The mechanization of the navigation equations uses updating of the transformation from the body frame to the computational frame. The body frame orientation is determined by aircraft attitude and high speed computations are required to update the body-to-computational frame transformation in a maneuvering vehicle.

For the space-stabilized computational frame, the body-to-inertial frame transformation may be updated at a higher rate than the transformation from the inertial frame to the local vertical north frame, required for the navigation computations. For the local-level computational frame,however, the total transformation from the body frame to the local-level frame must be updated at a high rate.

The transformation from the body frame to the computational frame may be expressed by a direction cosine matrix or by a quaternion. The quaternion representation of the transformation is more commonly employed in the present strapdown (SD) inertial navigation systems.

This subsection presents mechanizations for a strapdown system with a local-level-wander-azimuth computational frame which update the entire body-to-computational frame transformation in the high speed loop.

2.3.3.1 Strapdown Computations with Local Vertical Wander Azimuth Computational Frame

The (corrected) gyro outputs in a strapdown system represent the integral of the angular rate (and derivatives) of the sensor block with respect to an inertial reference frame expressed in sensor (body) coordinates over the sampling interval, $\underline{\Delta\theta}^b_{ib}$. If a measure of the integral of the angular rate of the computational frame with respect to the inertial frame expressed in body coordinates, $\underline{\Delta\theta}^b_{ic}$, is available, then the difference between the two incremental angles approximates the integral of the angular rate of the body frame with respect to the computational frame in body frame coordinates, $\underline{\Delta\theta}^b_{cb}$.

Expressing the angular rates in vector form:

$$\begin{aligned} \underline{\omega}^b_{cb} &= \underline{\omega}^b_{ib} + \underline{\omega}^b_{ci} \\ &= \underline{\omega}^b_{ib} - \underline{\omega}^b_{ic} \\ &= \underline{\omega}^b_{ib} - C^b_c \ \underline{\omega}^c_{ic} \end{aligned}$$

or $$\underline{\omega}^b_{cb} = \underline{\omega}^b_{ib} - (C^c_b)^T \ \underline{\omega}^c_{ic} \qquad (2\text{-}1)$$

Equation 2-1 is the instantaneous angular velocity of the body frame with respect to the computational frame in body coordinate. The quaternion differential equation to be integrated is:

$$\dot{q}^c_b = 1/2 \ q^c_b \ \underline{\omega}^b_{cb} \qquad (2\text{-}2)$$

The corresponding direction cosine matrix differential equation is:

$$\dot{C}^c_b = C^c_b \ \Omega^b_{cb} \qquad (2\text{-}3)$$

where Ω^b_{cb} is the three by three skew symmetric matrix corresponding to $\underline{\omega}^b_{cb}$, that is

$$\Omega^b_{cb} \triangleq \begin{bmatrix} 0 & -\omega^b_{cb_z} & \omega^b_{cb_y} \\ \omega^b_{cb_z} & 0 & -\omega^b_{cb_x} \\ -\omega^b_{cb_y} & \omega^b_{cb_x} & 0 \end{bmatrix} \sim \begin{bmatrix} \omega^b_{cb_x} \\ \omega^b_{cb_y} \\ \omega^b_{cb_z} \end{bmatrix} \cdot \equiv \underline{\omega}^b_{cb} \qquad (2\text{-}4)$$

It is necessary to integrate either Equation 2-2 or Equation 2-3 to update the body-to-computational frame transformation, C^c_b, which is required to transform the corrected accelerometer outputs from the body frame to the computational frame. Of course, the transpose of C^c_b is also required to transform $\underline{\omega}^c_{ic}$ from the computational to the body frame as indicated in Equation 2-1. When Equation 2-3 is to be used, Equations 2-1 and 2-4 are applied in that order to obtain Equation C-3.

In all SD systems examined to date, $\underline{\omega}^c_{ic}$ is computed, once per navigation cycle, Δt_n, from

$$\underline{\omega}^c_{ic} = \underline{\omega}^c_{ec} + \underline{\omega}^c_{ie} \qquad (2\text{-}5)$$

and it is implicitly assumed that $\dot{\underline{\omega}}^c_{ic}$ is zero over Δt_n which is tantamount to stating that

$$\dot{\underline{\omega}}^b_{ic} = \dot{C}^b_c \, \underline{\omega}^c_{ic} \qquad (2\text{-}6)$$

The difference equation corresponding to the integral of Equation 2-1 over the high speed computation cycle, Δt_n, is usually expressed as:

$$\underline{\Delta\theta}^{b}_{cb}\left(t_{n-1}+N\Delta t_N\right)=\underline{\Delta\theta}^{b}_{ib}\left(t_{n-1}+N\Delta t_N\right)-\left(C^{c}_{b}\left[t_{n-1}+(N-1)\Delta t_N\right]\right)^{T}\ \underline{\omega}^{c}_{ic}\left(t_{n-1}\right)\Delta t_N \qquad (2-7)$$

where:

$\Delta t_N = \Delta t_n / K$

$K = 2^m$, m is positive integer

Δt_n is the navigation computation cycle

$1 \leq N \leq K$

$t_{n-1} = (n-1)\Delta t_n$

$t_n = t_{n-1} + K\Delta t_N$

The only, computational difference between the high speed loop for a strapdown system with a SS computational frame and the same system with a LVWA computational frame is the additional computations given by Equation 2-1. The additional computation adds 9 multiplications and 9 additions per high speed computation cycle. The computational flow diagram for a strapdown (SD) system with LVWA navigation computations is shown in Figure 2-3. The updating of directional cosine matrix, shown in Figure 2-3, may be replaced by quaternion updating. The quaternion and direction cosine matrix updates are discussed in section 2.3.3.2.

2.3.3.2 Direction Cosine and Quaternion Update Algorithm

The equations used in generating the update algorithms are extracted from Reference (16), Chapter III. The update algorithms are approximations to integrals of Equations 2-2 or 2-3. If Equation 2-2 is integrated, the rotation quaternion must be converted to a rotation matrix at least for extraction of attitude angles, an operation requiring at least 9 multiplications and 12 additions. In either case the quaternion (or matrix) must be periodically normalized (or orthonormalized) to ensure that a rotation is truly represented. A first order algorithm is invariably employed for this purpose. A quaternion normalization requires 8 multiplications and 4 or 5 additions, while a matrix orthonormalization requires 54 multiplications and 42 additions plus, in either case, a number of loads, stores, shifts, etc. This latter operation constitutes one of the chief attractions of the quaternion update.

2.3.3.2.1 Direction Cosine Updating

For the case of the rotation (direction cosine), matrix update of order i = 1, 2, 3, the updated matrix is $C_i(t + \Delta t)$ where

$$C_i(t + \Delta t) = C(t)\, M_i(t,\Delta t) \tag{2-8}$$

and

$$M_1(t,\Delta t) = I + \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \tag{2-9}$$

and

$$M_2(t,\Delta t) = M_1(t,\Delta t) + \frac{1}{2}\begin{bmatrix} -\theta_y^2-\theta_z^2 & \theta_x\theta_y & \theta_z\theta_x \\ \theta_x\theta_y & -\theta_z^2-\theta_x^2 & \theta_y\theta_z \\ \theta_z\theta_x & \theta_y\theta_z & -\theta_x^2-\theta_y^2 \end{bmatrix} \tag{2-10}$$

and $M_3(t,\Delta t) = M_2(t,\Delta t)$

$$+\frac{1}{12}\begin{bmatrix} 0 & 2\theta^2\theta_z-\theta_y\theta_x^*+\theta_x\theta_y^* & -2\theta^2\theta_y-\theta_z\theta_x^*+\theta_x\theta_z^* \\ -2\theta^2\theta_z+\theta_y\theta_x^*-\theta_x\theta_y^* & 0 & 2\theta^2\theta_x-\theta_z\theta_y^*+\theta_y\theta_z^* \\ 2\theta^2\theta_y+\theta_z\theta_x^*-\theta_x\theta_z^* & -2\theta^2\theta_x+\theta_z\theta_y^*-\theta_y\theta_z^* & 0 \end{bmatrix} \quad (2\text{-}11)$$

In the foregoing, θ_j, where j = x, y, z, was used instead of the components of $\underline{\Delta\theta}^b_{cb}$ (or $\underline{\Delta\theta}^b_{ib}$) simply for economy of notation. θ_j^* is θ_j from the previous high speed computation cycle.

$$\theta^2 = \theta_x^2 + \theta_y^2 + \theta_z^2$$

It is noted from Equations 2-9 and 2-11 that the off-diagonal elements of the first and third order update matrix form a skew symmetric matrix; hence, there are only three distinct elements to compute. The second order update matrix is symmetric and involves six distinct elements. Thus a subroutine may easily be constructed to perform a first, second, or third order update depending only on the value of a control parameter, IC, where IC = IOU-2, and IOU is the integer order of the update. The flow chart of directional cosine updating is shown in Figure 2-21 .

The computations for a third order update are divided into four boxes 10, 20, 30, and 40 in the flow chart, Figure 2-1. For a second order update only the computations in boxes 10, 30 and 40 are performed. For a first order update only box 30 is required. Following the indicated operations, the body-to-computational frame rotation matrix is updated (and periodically orthonormalized). Then the ΔV's are transformed and summed over Δt_n.

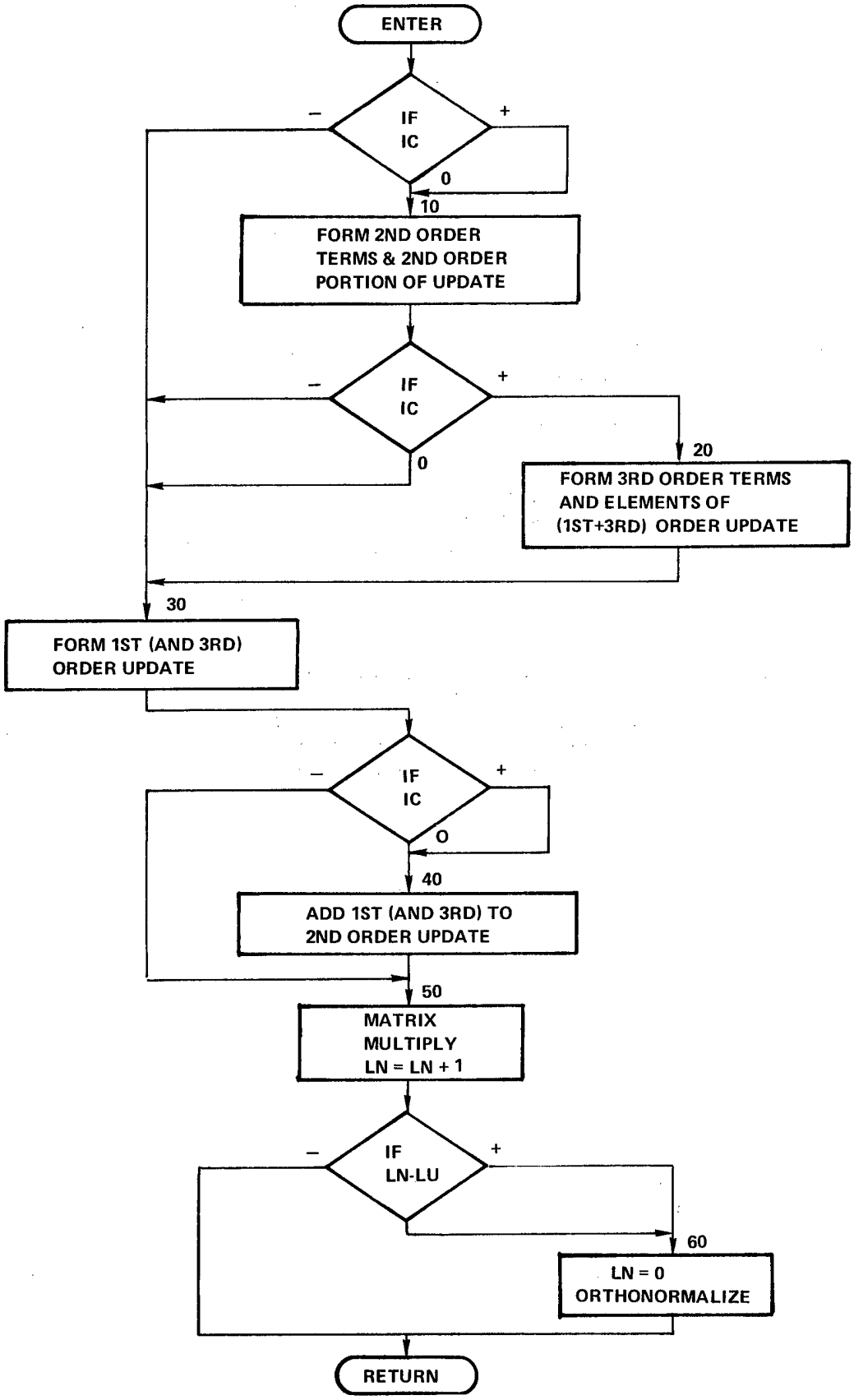


Figure 2-7. Flow Chart for 1st, 2nd or 3rd Order Direction Cosine Update Matrix

BOX 10 – FORM SECOND ORDER TERMS AND SECOND ORDER PORTION OF UPDATE

DTXS ⇐ (DTHX*DTHX)/2
DTYS ⇐ (DTHY*DTHY)/2
DTZS ⇐ (DTHZ*DTHZ)/2
DTXY ⇐ (DTHX*DTHY)/2
DTYZ ⇐ (DTHY*DTHZ)/2
DTZX ⇐ (DTHZ*DTHX)/2

$$[DA2] \Leftarrow \begin{bmatrix} -DTYS-DTZS & DTXY & DTZX \\ DTXY & -DTZS-DTXS & DTYZ \\ DTZX & DTYZ & -DTXS-DTYS \end{bmatrix}$$

BOX 20 – FORM THIRD ORDER TERMS AND ELEMENTS OF FIRST PLUS THIRD ORDER UPDATE

XDTX ⇐ (DTHZ*ODTY–DTHY*ODTZ)*(1/12)
XDTY ⇐ (DTHX*ODTZ–DTHZ*ODTX)*(1/12)
XDTZ ⇐ (DTHY*ODTX–DTHX*ODTY)*(1/12)

Figure 2-7a. Updating of Body-to-Computation Frame Direction Cosines Matrix

BOX 20 (CONTINUED) –

ODTX ⇐DTHX
ODTY ⇐DTHY
ODTZ ⇐DTHZ

SCL1 ⇐1. – (DTXS+DTYS+DTZS)*(1/3)

DTHX ⇐SCL1*DTHX+DTX
DTHY ⇐SCL1*DTHY+XDTY
DTHZ ⇐SCL1*DTHZ+XDTZ

BOX 30 – FORM FIRST PLUS THIRD ORDER PORTION OF UPDATE

$$[R5.5N] \Leftarrow \begin{bmatrix} 1. & -DTHZ & DTHY \\ DTHZ & 1. & -DTHX \\ -DTHY & DTHX & 1. \end{bmatrix}$$

BOX 40 – ADD FIRST PLUS THIRD ORDER PORTION TO SECOND ORDER PORTION

$$[R5.5N] \Leftarrow [R5.5N] + [DA2]$$

Figure 2-7b. Direction Cosine Matrix Updating (Continued)

2.3.3.2.2 Quarternion Updating

The steps involved in generating an updated rotation matrix relating the computational frame to the body frame, given the $\underline{\Delta\theta}^{b}_{cb}$, are now presented using quaternions. As before first, second, and third order updates are considered. In the present case the updated quaternion is $q^{(i)}(t+\Delta t)$ where

$$q_i^{(i)}(t+\Delta t) = q(t)\ p^{(i)}(\Delta t) \tag{2-12}$$

and $$p^{(1)}(\Delta t) = \left(1 + \frac{\underline{\theta}}{2}\right) \tag{2-13}$$

and $$p^{(2)}(\Delta t) = p^{(1)}(\Delta t) - \frac{1}{2}\frac{\underline{\theta}}{2} \cdot \frac{\underline{\theta}}{2} \tag{2-14}$$

and $$p^{(3)}(\Delta t) = p^{(2)}(\Delta t) - \frac{1}{6}\left(\frac{\underline{\theta}}{2} \cdot \frac{\underline{\theta}}{2}\right)\frac{\underline{\theta}}{2} + \frac{1}{6}\left(\frac{\underline{\theta}}{2}^{*} \times \frac{\underline{\theta}}{2}\right) \tag{2-15}$$

The operation indicated in Equation C-12 is multiplication of two quaternions where:

$$p \triangleq p_0 + \underline{p}$$

and $$\underline{p} \triangleq p_1\hat{i} + p_2\hat{j} + p_2\hat{k}$$

and $\hat{i}$, $\hat{j}$, $\hat{k}$ are the frame vectors (unit magnitude, mutually orthogonal, right handed) (in the particular case $\hat{i}$, $\hat{j}$, $\hat{k}$ are along the x, y, z axes of the body frame).

A good concise summary of the elements of quaternion algebra is presented in Section 2.2.1 of Reference 16. The product of two quaternions, p and q, may be written as

$$qp = q_0p_0 - \underline{q}\cdot\underline{p} + q_0\underline{p} + \underline{q}p_0 + \underline{q}\times\underline{p}$$

The preceding vector-scalar form may be handy analytically but the most efficient form for mechanization in a digital computer is as the product of a 4 x 4 matrix by a 4 x 1 vector, i.e.,

$$qp = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & p_3 & -p_2 \\ p_2 & -p_3 & p_0 & p_1 \\ p_3 & p_2 & -p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_o \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (2\text{-}16)$$

The purist may object that Equation 2-16 is an expression for the elements of the quaternion product rather than the product per se which, of course, is correct. To be rigorous then the left hand side of Equation 2-16 should be expressed as

$$\begin{bmatrix} (qp)_0 & (qp)_1 & (qp)_2 & (qp)_3 \end{bmatrix}^T .$$

The rearranged product pq is given below:

$$pq = p_0q_0 - \underline{p} \cdot \underline{q} + p_0\underline{q} + \underline{p}q_0 + \underline{p}\times\underline{q} \quad (2\text{-}17)$$

$$\begin{bmatrix} (pq)_0 \\ (pq)_1 \\ (pq)_2 \\ (pq)_3 \end{bmatrix} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

The first and third order updates affect the elements p_1, p_2, and p_3, while the second order update portion affects only the element p_0. All orders of quaternion updates work with half the angle rotation, hence the first operation is to divide the elements of the corrected $\underline{\Delta\theta}^b$ by two (right shift) and work thereafter with $\underline{\Delta\theta}^b/2$.

For the first order update, the quaternion elements, $p_i^{(1)}$, are

$$\begin{aligned} p_0^{(1)} &= 1 \\ p_1^{(1)} &= \theta_1/2 \\ p_2^{(1)} &= \theta_2/2 \\ p_3^{(1)} &= \theta_3/2 \end{aligned} \tag{2-18}$$

For the second order update, $p_0^{(2)}$ becomes

$$p_0^{(2)} = 1 - \frac{1}{2}\left(\frac{\theta}{2} \cdot \frac{\theta}{2}\right) = 1 - \frac{1}{2}\left(\frac{\theta}{2}\right)^2 = 1 - \frac{1}{2}\left[\left(\frac{\theta_1}{2}\right)^2 + \left(\frac{\theta_2}{2}\right)^2 + \left(\frac{\theta_3}{2}\right)^2\right] \tag{2-19}$$

For the third order update, $p_1^{(3)}$, $p_2^{(3)}$, $p_3^{(3)}$, become

$$\begin{aligned} p_1^{(3)} &= \left[1 - \frac{1}{6}\left(\frac{\theta}{2} \cdot \frac{\theta}{2}\right)\right]\frac{\theta_1}{2} + \frac{1}{6}\left(\frac{\theta}{2}^* \times \frac{\theta}{2}\right) \\ &= \left[1 - \frac{1}{6}\left(\frac{\theta}{2}\right)^2\right]\frac{\theta_1}{2} + \frac{1}{6}\left(\frac{\theta_3}{2}\frac{\theta_2^*}{2} - \frac{\theta_2}{2}\frac{\theta_3^*}{2}\right) \\ p_2^{(3)} &= \left[1 - \frac{1}{6}\left(\frac{\theta}{2}\right)^2\right]\frac{\theta_2}{2} + \frac{1}{6}\left(\frac{\theta_1}{2}\frac{\theta_3^*}{2} - \frac{\theta_3}{2}\frac{\theta_1^*}{2}\right) \\ p_3^{(3)} &= \left[1 - \frac{1}{6}\left(\frac{\theta}{2}\right)^2\right]\frac{\theta_3}{2} + \frac{1}{6}\left(\frac{\theta_1}{2}\frac{\theta_3^*}{2} - \frac{\theta_1}{2}\frac{\theta_2^*}{2}\right) \end{aligned} \tag{2-20}$$

Note that the subscripts 1, 2, 3 are equivalent to subscripts x, y, z, used in the direction cosine update.

A similar logical flow may be employed for the elements of the update quaternion as was used in the direction cosine update. The initial step, as noted previously, is division of the $\underline{\Delta\theta}^b$ by two. Following generation of the $p_i^{(j)}$, the 4 x 4 matrix is formed and the quaternion multiplication is performed (4 x 4) x (4 x 1) with subsequent normalization (perhaps at lower frequency) and conversion of the updated quaternion to a direction cosine matrix.

The ensuing $\underline{\Delta v}$ transformation and summation and attitude extraction are, of course, independent of the method used to construct the direction cosine matrix.

The flow chart shown in Figure 2-8 presents the computations starting with $\underline{\Delta\theta}^b$ and ending with the updated C_b^c (or C_b^i). The details of computations are shown in Figures 2-8a, b and c.

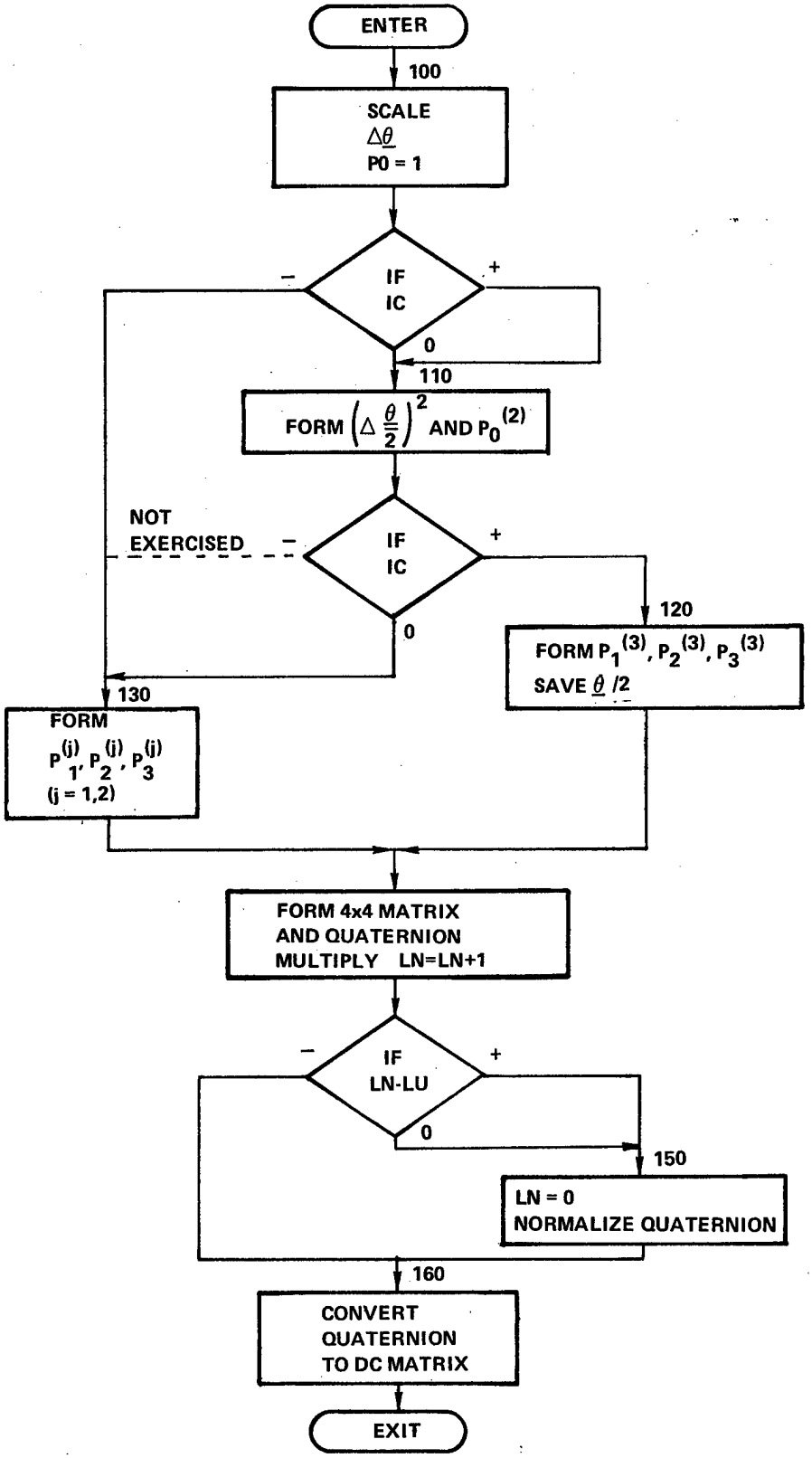


Figure 2-8. Flow Chart for 1st, 2nd, and 3rd Order Quaternion Update

BOX 100 – SCALE $\Delta\theta_i$'s AND SET $p_0 = 1$

DTHX ⇐DTHX/2
DTHY ⇐DTHY/2
DTHZ ⇐DTHZ/2

P0 ⇐1.

BOX 110 – FORM $\left(\frac{\Delta\theta}{2}\right)^2$ AND po FOR SECOND ORDER

TSO2 ⇐(DTHX*DTHX+DTHY*DTHY+DTHZ*DTHZ)/2
PO ⇐1. – TSO2

BOX 120 – FORM 3RD ORDER TERMS, SAVE $\Delta\theta_i$'s.

XDTX ⇐(DTHZ*ODTY–DTHY*ODTZ)*(1/6)
XDTY ⇐(DTHX*ODTZ–DTHZ*ODTX)*(1/6)
XDTZ ⇐(DTHY*ODTX–DTHX*ODTY)*(1/6)
CON ⇐1. – TSO2*(1/3)

P1 ⇐CON*DTHX + XDTX
P2 ⇐CON*DTHY + DTY
P3 ⇐CON*DTHZ + XDTZ

Figure 2-8a. Updating of Body-to-Computational Frame, Direction Cosine Matrix, Using Quaternion Transformation

BOX 120 – ODTX ⇐DTHX
ODTY ⇐DTHY
ODTZ ⇐DTHZ

BOX 130 – FORM 1ST OR 2ND ORDER P_1, P_2, P_3
P1 ⇐DTHX
P2 ⇐DTHY
P3 ⇐DTHZ

BOX 140 – FORM UPDATE MATRIX AND MULTIPLY

$$[UPDQ] \triangleq \begin{bmatrix} P0 & -P1 & -P2 & -P3 \\ P1 & P0 & P3 & -P2 \\ P2 & -P3 & P0 & P1 \\ P3 & P2 & -P1 & P0 \end{bmatrix}$$

$$\begin{bmatrix} Q0 \\ Q1 \\ Q2 \\ Q3 \end{bmatrix} \Leftarrow [UPDQ]^* \begin{bmatrix} Q0 \\ Q1 \\ Q2 \\ Q3 \end{bmatrix}$$

Figure 2-8b. Computation for Updated D.C.M. (continued)

BOX 150 – NORMALIZE UPDATED QUATERNION

COR ⇐1. – (Q0*Q0+Q1*Q1+Q2*Q2+Q3*Q3–1.)/2

$$\begin{bmatrix} Q0 \\ Q1 \\ Q2 \\ Q3 \end{bmatrix} \Leftarrow COR * \begin{bmatrix} Q0 \\ Q1 \\ Q2 \\ Q3 \end{bmatrix}$$

BOX 160 – CONVERT QUATERNION INTO DC MATRIX (Cij)

Q1S ⇐Q1*Q1
Q2S ⇐Q2*Q2
Q3S ⇐Q3*Q3
Q01 ⇐Q0*Q1
Q02 ⇐Q0*Q2
Q03 ⇐Q0*Q3
Q12 ⇐Q1*Q2
Q23 ⇐Q2*Q3
Q31 ⇐Q3*Q1

C11 ⇐1. – 2*(Q2S+Q3S)
C12 ⇐2*(Q12–Q03)
C13 ⇐2*(Q31+Q02)
C21 ⇐2*(Q12+Q03)
C22 ⇐1. – 2*(Q3S+Q1S)
C23 ⇐2*(Q23–Q01)
C31 ⇐2*(Q31–Q02)
C32 ⇐2*(Q23+Q01)
C33 ⇐1.–2*(Q1S+Q2S)

Figure 2-8c. Computation for Updated D.C.M. (continued)

2.3.3.3 Δv Transformation for SDIMU

For the SD systems considered both the incremental angles, $\underline{\Delta\theta}^b_{ib}$, and the incremental velocities, $\underline{\Delta v}^b_i$ are accumulated over the same interval, Δt_N. This is compatible with the output format of the profile generator program. Some SD systems alternate the outputs of $\underline{\Delta\theta}^b_{ib}$ and $\underline{\Delta v}^b$ at intervals of $\Delta t_{N/2}$. While the updated C^c_b is appropriate for attitude computations, it is one half a high speed computation cycle ($\Delta t_{N/2}$) ahead of the transformation required for the $\underline{\Delta v}^b$. An interpolated value for C^c_b is required for $\underline{\Delta v}^b$ transformation. When a direction matrix update is used, it would feasible to employ techniques analogous to those described in section 2.4.4 for extrapolating the direction cosine matrix in the navigation algorithm. However, when a quaternion update is used, there is nothing corresponding to the "update" matrix which is added to the "old" direction cosine matrix to obtain the "new" direction cosine matrix (dcm).

The simple expedient employed in both cases was simply to store the "old" dcm, $C^c_{b_{N-1}}$, before generating the "new" dcm, $C^c_{b_N}$, and then after the update use the average matrix as $C^c_{b_{N-1/2}}$ for the $\underline{\Delta v}^b$ transformation.
i.e.,

$$C^c_{b_{N-1/2}} = \frac{1}{2}\left(C^c_{b_{N-1}} + C^c_{b_N}\right) \tag{2-21}$$

While the above equation produces errors which are second order in $\underline{\Delta\theta}$, (see Appendix E), they are not cumulative in their effect on C^c_b.

Finally, the transformed Δv^b is given by

$$\underline{\Delta v}^c_N = C^c_{b_{N-1/2}}\ \underline{\Delta v}^b_N \tag{2-22}$$

In the foregoing all the subscripts, N, N-1/2, N-1, refer to the high speed strapdown computation cycle.

2.4 Navigation Equations

2.4.1 Navigation Equations in Local Vertical Frame

The fundamental equation of inertial navigation for a navigator employing an earth relative computational frame (c-space) is given by

$$\dot{\underline{v}}^c = \underline{f}^c + \underline{g}^c - (\Omega^c_{ec} + 2\Omega^c_{ie})\underline{v}^c \qquad (2\text{-}23)$$

where $\underline{v}^c$ is the earth relative velocity in the computational frame

$\dot{\underline{v}}^c$ is the time rate of change of $\underline{v}^c$

$\underline{f}^c$ is the specific force (the portion of inertial acceleration sensed by the accelerometers) expressed in computational frame coordinates

$\underline{g}^c \triangleq \underline{G}^c - \Omega^c_{ie}\,\Omega^c_{ie}\,\underline{r}^c$, is the resultant of mass attraction, $\underline{G}^c$, and centripetal force, $-\Omega^c_{ie}\,\Omega^c_{ie}\,\underline{r}^c$, expressed in computational frame coordinates

e is the earth fixed frame which is rotating at a constant angular rate, ω_{ie}, with respect to the inertial frame

i is the earth-centered inertial frame which is regarded as non rotating with respect to the "fixed" stars

Ω^c_{ec} is the skew-symmetric form of the angular velocity of the computational frame with respect to the earth fixed frame, expressed in computational frame coordinates, $\underline{\omega}^c_{ec}$.

Ω^c_{ie} is the skew-symmetric form of the angular velocity of the earth-fixed frame with respect to the inertial frame, expressed in computational frame coordinates, $\underline{\omega}^c_{ie}$. The magnitude of $\underline{\omega}^c_{ie}$, ω_{ie}, is a constant but the components vary as a function of latitude (and wander angle).

The function of the navigation computer may be considered to be the integration of equation (2-23), to obtain the earth relative velocity, $\underline{v}^c$, followed by a second integration to obtain

the earth relative position. Since the earth relative position is expressed mainly in angular coordinates, latitude, longitude, (and wander angle), it is necessary to convert the level components of linear velocity from the first integration into angular velocity form by dividing by the appropriate variable, radii of curvature to obtain $\underline{\omega}^{c}_{ec}$ which is then converted to the skew symmetric form, Ω^{c}_{ec}. The differential equation which is integrated to obtain the angular position is

$$\dot{C}^{e}_{c} = C^{e}_{c}\ \Omega^{c}_{ec} \tag{2-24}$$

The actual latitude, longitude (and wander angle) are extracted from C^{e}_{c} via the appropriate inverse trigonometric functions.

The remaining position variable, altitude is obtained by integrating the vertical component of $\underline{v}^{c}$. An external altitude reference is required to eliminate the inherent instability of any pure inertial vertical channel. Thus, the minimum number of scalar integrations required to mechanize a local vertical wander azimuth navigator is thirteen (13). However, only five of the nine (9) elements of C^{e}_{c} are employed directly in the navigation computations so frequently only six of the nine elements are computed.

2.4.1.1 Methods of Presentation

There are a number of ways of presenting the equations which are to be mechanized in a navigation computer.

(a) The highest level presents the exact differential equations (and other equations or functions for angular velocity, gravity, etc.). At this level, there is no order to the equations since all the values of the variables and their derivatives are instantaneous. The same information may be expressed in state space (matrix-vector) form or in analog block diagram form.

(b) A second level is the difference equation form, which is probably the most informative quasi-analytical method. Here, too, the order in which the equations appear is irrelevant since all the variables are tagged with the (discrete) times when they were computed. From the difference equations, detailed computational flow diagrams may be constructed which convey the same information in a more graphic form. This latter form implies a particular sequence or order to the computations.

(c) A third level is the functional flow chart which generally contains a verbal description of an unalterable sequence of arithmetic operations on a number of variables and constants within a single "box". Logic and/or decisions maybe incorporated via standard flowchart symbols. The order of the computations appears explicitly. The individual "boxes" may be inline coding, subroutines or functions.

(d) A fourth level is the detailed flowchart which may be manually drawn or from AUTOFLOW or similar program.

(e) A fifth level is the program listing in a higher order language (FORTRAN in this case). The usefulness of the listing is strongly dependent on the definition of the mnemonics and the information provided by the comments.

(f) The sixth (or lowest) level is the assembly language listing. In the present study this is of immediate interest only in the case of the precision reduction routine, where the routine was actually written in assembly language.

Generally, presentation at the third or lower level is incorporated in Volume III; the first and second levels are included in this volume. Derivations leading to the first or second levels are relegated to Appendices B, D, and E of Volume II.

2.4.1.2 Exact Equations

The equations to be integrated and the intervening computations, in exact form, are presented below for the undamped vertical case.

(1) Derivative of velocity

$$\dot{\underline{v}}^c = \underline{f}^c + \underline{g}^c - (\Omega^c_{ec} + 2\Omega^c_{ie})\underline{v}^c \tag{2-25}$$

(2) Level Components of Angular Velocity

$$\underline{v}^\ell = C^\ell_c \, \underline{v}^c \tag{2-26}$$

where $\underline{v}^\ell = \{v_u v_E v_N\}^T$ are the components of the earth relative velocities in the local vertical, path pointing frame (LVN).

$C^\ell_c = X(\alpha)$ is the transformation from the (LVWA) computational frame to the LVN frame.

$$\omega_E = \frac{-v_N}{r_m(L) + h} \tag{2-27}$$

$$\omega_N = \frac{v_E}{r_p(L) + h} \tag{2-28}$$

$$\text{where } r_m(L) = \frac{r_o(1-\varepsilon^2)}{(1-\varepsilon^2 \sin^2 L)^{1/2}} \tag{2-29}$$

$$\text{and } r_p(L) = \frac{r_o}{(1-\varepsilon^2 \sin^2 L)^{1/2}} \tag{2-30}$$

$r_m(L)$, $r_p(L)$ are the meridian and prime vertical radii of curvature on the surface of the ellipsoid at geodetic latitude, L = Lg, defined in Appendix C.

r_o = a, is the sealevel equatorial radius of the earth (or the semi major axis of the meridional ellipse).

$\varepsilon = \left(1 - \frac{b^2}{a^2}\right)^{1/2}$ is the eccentricity of the meridional ellipse where b is the polar radius (or semi-minor axis).

ω_E, ω_N are the level components of the angular velocity of the

vehicle with respect to the earth in the LVN frame, $\underline{\omega}^{\ell}_{e\ell}$

$$\underline{\rho} = C^{c}_{\ell} \{0 \ \omega_e \ \omega_N\}^T \tag{2-31}$$

where $\underline{\rho} \triangleq \underline{\omega}^{c}_{ec}$ is the angular velocity of the computation frame with respect to the earth fixed frame in the computational frame.

(3) Derivative of Direction Cosine Matrix

$$\dot{C}^{e}_{c} = C^{e}_{c} \ \Omega^{c}_{ec} \tag{2-32}$$

$$\text{where } \Omega^{c}_{ec} = \begin{bmatrix} 0 & -\rho_z & \rho_y \\ \rho_z & 0 & -\rho_x \\ -\rho_y & \rho_x & 0 \end{bmatrix} \tag{2-33}$$

(4) Earth's Rate and Gyro Torquing

$$\underline{\omega}^{c}_{ic} = \underline{\omega}^{c}_{ec} + \underline{\omega}^{c}_{ie} = \underline{\rho} + \underline{\omega}^{c}_{ie} \tag{2-34}$$

$$\text{where } \underline{\omega}^{e}_{ie} = \underline{\omega}^{i}_{ie} = \{0 \ 0 \ \omega_{ie}\}^T \tag{2-35}$$

$$\text{and } \underline{\omega}^{c}_{ie} = (C^{e}_{c})^T \underline{\omega}^{e}_{ie} \tag{2-36}$$

(5) Coriolis and Centripetal Acceleration

$$(\underline{\omega}^{c}_{ec} + 2 \ \underline{\omega}^{c}_{ie}) \times \underline{v}^{c} = (\underline{\omega}^{c}_{ic} + \underline{\omega}^{c}_{ie}) \times \underline{v}^{c} \tag{2-37}$$

$$= (\Omega^{c}_{ic} + \Omega^{c}_{ie}) \ \underline{v}^{c}$$

where Ω^{c}_{ic} and Ω^{c}_{ie} are the skew-symmetric forms of $\underline{\omega}^{c}_{ic}$, the gyro torquing signals, and $\underline{\omega}^{c}_{ie}$, the earth rate in the computational frame.

(6) Gravity Computation

$$\underline{g}^c = C_\ell^c \, \underline{g}^\ell$$

$$\text{where } \underline{g}^\ell = \begin{bmatrix} -(g_0 + g_2 \sin^2 L + g_4 \sin^4 L)[1 - (g_{h_1} - g_{h_2} \sin^2 L)h + g_{h_3} h^2] \\ 0 \\ -g_N h \sin L \cos L \end{bmatrix} \tag{2-39}$$

and g_0, g_2, g_4, g_{h_1}, g_{h_2}, g_{h_3}, and g_N are constants in the gravity equation

h is the altitude

sinL, cosL are the sine and cosine of the geodetic latitude

(7) Derivative of altitude

$$\dot{h} = \dot{v}_x^c \tag{2-40}$$

(8) Position extraction and LVN Velocities

$$L = \tan^{-1} \left\{ \frac{(C_c^e)_{31}}{\left[(C_c^e)_{11}^2 + (C_c^e)_{21}^2 \right]^{1/2}} \right\} \tag{2-41}$$

$$\lambda = \tan^{-1} \left[\frac{(C_c^e)_{21}}{(C_c^e)_{11}} \right] + \lambda_o \tag{2-42}$$

$$\alpha = \tan^{-1} \left[\frac{-(C_c^e)_{32}}{(C_c^e)_{33}} \right] \tag{2-43}$$

$$\underline{v}^\ell = C_c^\ell \, \underline{v}^c \tag{2-44}$$

$$\text{where } C_c^\ell = X(\alpha) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \tag{2-45}$$

and α is the present wander angle (positive east of north)

λ, λ_o are the present and initial longitude (positive east of Greenwich meridian)

L is the geodetic latitude (positive north of the equator)

$\underline{v}^{\ell} = \{v_U v_E v_N\}$ is the LVN velocity

2.4.2 Vertical Channel Damping

The inherent instability of the vertical channel of any pure inertial navigator is due to errors in the computation of $\underline{g}^c$ resulting from errors in the computed altitude or due to vertical accelerometer errors. Linearizing the expression for the vertical component of gravity in the previous section we have

$$g_x^\ell = g_x^c \approx -g_o\,(1-C_{h_1} h) \tag{2-46}$$

where $C_{h_1} \approx \frac{2}{r_o}$

or

$$g_x^c \approx -g_o + \frac{2g_o}{r_o} h \tag{2-47}$$

The second term on the right is the potential positive feedback term and will result in an exponentially diverging altitude error for the most infinitesimal initial error. A block diagram of the undamped vertical channel is show in Figure 2-9.

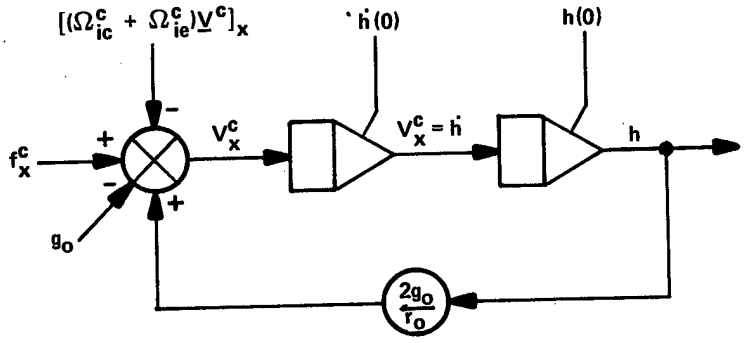


Figure 2-9. Undamped Vertical Channel Block Diagram

A non-inertial, external altitude reference, h_B (generally a barometric altimeter) is incorporated into the vertical channel via conditional feedback loops as illustrated in Figure 2-10

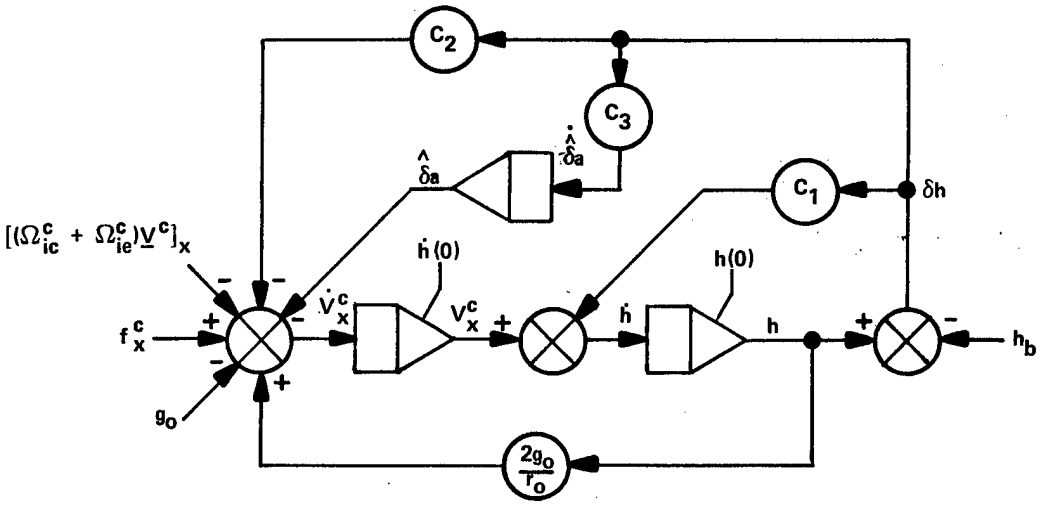


Figure 2-10. Third Order Damped Vertical Block Diagram

The characteristic equation for loop shown in Figure 2-10.

$$s^3 + C_1 s^2 + \left(C_2 - \frac{2g_o}{r_o}\right) s + C_3 = 0 \qquad (2\text{-}48)$$

which represents a stable damped, third order system for all of C_1, C_2, C_3 positive and

$$C_2 > \frac{2g_o}{r_o} . \qquad (2\text{-}49)$$

If $C_3 = 0$, the system reduces to second order. Both cases, with a range of values for the parameters, occur in practice.

The additional (differential) equations which are required for third order damping implementation are:

$$1)\quad \dot{v}^c_x = f^c_x + g^c_x - [(\Omega^c_{ec} + \Omega^c_{ic})\underline{v}^c]_x \underline{-C_2\delta h - \hat{\delta}a}$$

$$2)\quad \dot{h} = v^c_x \underline{- C_1\ \delta h} \qquad (2\text{-}50)$$

$$3)\quad \underline{\dot{\hat{\delta}}a = C_3\ \delta h}$$

$$4)\quad \underline{\delta h = h - h_B}$$

Typed underlining indicates the additions.

2.4.3 "Baseline" LVWA Navigation Algorithm (in Difference Equation Form

The equations mechanized in the "baseline" LVWA Navigation Algorithm are ordered as shown in Table 2-1. Each number represents a fixed sequence of operations applicable at all levels.

Table 2-1

Local Vertical Wander Azimuth Navigation Algorithm with Vertical Damping		
1	Update computational frame velocities. (store old velocities)	INRVAV
2	Add vertical damping terms.	
3	Compute angular velocities about level axes. INRNAV or ANGVL2	
4	Update direction cosine matrix (1st or 2nd order). DCMUPD	
5	Orthonormalize direction cosine matrix (1st order). ORTHO	
6	Compute earth's rate signals	TORCOR
*7	Compute gyro torquing signals.	
8	Compute angular velocity [for $(\underline{\omega} x \underline{v})$].	
9	Compute $(\underline{\omega} x \underline{v})\Delta t$ terms (for velocity update)	
10	Compute altitude and vertical damping. INRNAV	
11	Compute gravity components. GRAV	
*12	Extract position and wander angle.	POSVEL
*13	Form wander angle matrix and LVN velocities.	

*Shown as separate blocks in computational flow diagrams.

General

Second level consists of presenting the approximate integrals of the differential equations and approximation to the other functions in difference equation form, generally using a rectangular integration scheme.

Generally, one (or more) fixed navigation computation cycle(s) is (are) employed. Consider just one computation cycle for navigation, Δt, where Δt is a binary fraction of a second for minimization of the computational load and truncation. Time in navigation is set to zero (t=0) at the moment of entry into navigation i.e., at the conclusion of initialization; thereafter elasped time is updated by counting the number of navigation computation cycles which have elasped since initialization. Time at end of the $n^{\underline{th}}$ navigation computation cycle is denoted by t_n where $t_n = n\Delta t$.

The function of the navigation algorithm is to update the navigation parameters to t_n from their values at t_{n-1}, based on the incremental velocity and other data available at t_n. In the case of a local vertical system, the navigation computations at t_n must also provide gyro torquing signals to be applied from t_n to t_{n+1}.

NOTE: In passing to the difference equation form, the literal superscripts and subscripts designating coordinate frames are replaced by their numerical equivalents to correspond more closely with program mnemonics. The following table will clarify the designations:

Frame Name	Designation	
	Literal	Numerical
Inertial	i	0
Earth-fixed	e	0'
Local vertical, north pointing	ℓ	2
Local vertical, wander azimuth or computational	c	2'

1. Update Computational Frame Velocities

Defining the corrected incremental velocity output of the X accelerometer of a local vertical IMU at t_n as

$$\Delta v_x^{2'}(t_n) \triangleq \Delta v_{x_n}^{2'} = \int_{(n-1)\Delta t}^{n\Delta t} f_x^{2'}(\tau)d\tau - \varepsilon_x Q_x \qquad (2\text{-}51)$$

where Q_x is the quantization of the x-accelerometer and $-1<\varepsilon_x<1$, the velocity update equation for the simplest algorithm will be written as

$$\underline{v}_n^{2'}(u) = \underline{v}_{n-1}^{2'}(d) + \underline{\Delta v}_n^{2'} + \left[\underline{g}_n^{2'} - \left(\Omega_{0',2'}^{2'}{}_{n-1} + 2\Omega_{0,0'}^{2'}{}_{n-1}\right)\underline{v}_{n-1}^{2'}(d)\right]\Delta t \qquad (2\text{-}52)$$

where

$\underline{v}_n^{2'}(u)$ is the velocity vector in the 2' frame at time t_n undamped i.e. without the addition of the vertical damping correction.

$\underline{v}_{n-1}^{2'}(d)$ is the damped velocity vector in the 2' frame at time t_{n-1}.

$\Delta\underline{v}_n^{2'}$ is the vector, one component of which is given by Eqn (2-51).

$\underline{g}_{n-1}^{2'}$ is the gravity vector in the 2' frame computed at time t_{n-1} (or $(n-1)\Delta t$).

$\Omega_{0',2'}^{2'}{}_{n-1}$ is the skew symmetric form of the angular velocity of the 2' frame with respect to the 0'-frame expressed in the 2' frame and computed at time t_{n-1}, $\underline{\omega}_{0',2'}^{2'}{}_{n-1}$

$\Omega_{0,0'}^{2'}{}_{n-1}$ is the skew symmetric form of the angular velocity of the 0' frame with respect to the 0-frame expressed in the 2' frame and computed at time t_{n-1}.

Δt is the navigation computational cycle.

2. Add Vertical Damping Terms

Vertical damping terms are added for one component of velocity viz:

$$v_{x_n}^{2'}(d) = v_{x_n}^{2'}(u) + (C_2 \delta h_{n-1} + \hat{\delta a}_{n-1})\Delta t \qquad (2\text{-}53)$$

where the subscript x refers to the first or vertical component of velocity in the 2' frame

C_2 is a constant

$\delta h_{n-1} \triangleq h_{B_{n-1}} - h_{n-1}$

$h_{B_{n-1}}$ is the reference (barometric) altitude at t_{n-1}

h_{n-1} is the system computed altitude at t_{n-1}

$\hat{\delta a}_{n-1}$ is defined by Eqn. (2-67) with n replaced by n-1

3. Compute Angular Velocities About Level Axes

The level components of the angular velocity due to vehicle motion over the earth, $\underline{\omega}_{0',2'}^{2'}$, or $\underline{\rho}$ computed using the approximate equation are:

$$\omega_{y_{0',2'_n}}^{2'} = \frac{-v_{z_n}^{2'}}{r_o}\left\{1 - \frac{h_{n-1}}{r_o} - e\left[1 - 3\left(C_{2'_{n-1}}^{0'}\right)_{33}^2 - \left(C_{2'_{n-1}}^{0'}\right)_{32}^2\right]\right\}$$
$$- \frac{v_{y_n}^{2'}}{r_o} e \left(C_{2'_{n-1}}^{0'}\right)_{32}\left(C_{2'_{n-1}}^{0'}\right)_{33}$$

$$\omega_{z_{0',2'_n}}^{2'} = \frac{v_{y_n}^{2'}}{r_o}\left\{1 - \frac{h_{n-1}}{r_o} - e\left[1 - 3\left(C_{2'_{n-1}}^{0'}\right)_{32}^2 - \left(C_{2'_{n-1}}^{0'}\right)_{33}^2\right]\right\} \qquad (2\text{-}54)$$
$$+ \frac{v_{z_n}^{2'}}{r_o} e \left(C_{2'_{n-1}}^{0'}\right)_{32}\left(C_{2'_{n-1}}^{0'}\right)_{33}$$

NOTE: $\omega^{2'}_{x_{0',2'_n}} \triangleq 0$

where r_o is a constant equal to the equatorial sea level radius of the earth, a, i.e., the semi-major axis of the meridional ellipse; e is a constant equal to the ellipticity of the meridional ellipse, $\frac{a-b}{a}$, i.e. the difference between the semi-major and semi-minor axes of the meridional ellipse divided by the semi-major axis; h_{n-1} is the system computed altitude at time t_{n-1}.

$\left(C^{0'}_{2'_{n-1}}\right)_{32}$ is the second element of the third row of the LVWA to earth fixed transformation, $C^{0'}_{2'}$, at time t_{n-1}. It is equal to $-\cos L_{n-1} \sin\alpha_{n-1}$.

$\left(C^{0'}_{2'_{n-1}}\right)_{33}$ is the third element of the third row of the LVWA to earth fixed transformation, $C^{0'}_{2'}$, at time t_{n-1}. It is equal to $\cos L_{n-1} \cos\alpha_{n-1}$.

$\omega^{2'}_{i_{0',2'_n}}$ where i=x,y,z, is the i^{th} component of the angular velocity of the (LVWA) computational (2') frame with respect to the earth fixed (0') frame expressed in the 2' frame-computed at time t_{n-1}.

NOTE: (a) Due to the similarity in the expressions for the y and z components of $\underline{\omega}^{2'}_{0',2'_n}$, many computational simplifications can be realized in the actual mechanization of these expressions.

(b) For compactness of notation the elements of $\underline{\omega}^{2'}_{0',2'_n}$ are frequently expressed as ρ_{i_n} where i=x,y,z.

4. Update Direction Cosine Matrix

The direction cosine matrix, $C^{0'}_{2'_{n-1}}$, is updated (to the first order) using the ρ_{i_n} computed above viz:

$$C^{0'}_{2'_n} = C^{0'}_{2'_{n-1}} + C^{0'}_{2'_{n-1}} \Omega^{2'}_{0',2'_n} \Delta t \qquad (2\text{-}55)$$

where $\Omega^{2'}_{0',2'_n}$ is the skew symmetric form of $\underline{\omega}^{2'}_{0',2'_n}$ (or $\underline{\rho}_n$).

5. Orthonormalize Direction Cosine Matrix

Periodically, the direction cosine matrix, $C^{0'}_{2'_n}$, may be orthonormalized (to the first order) viz:

$$C^{0'}_{2'_n}(\text{ortho}) = C^{0'}_{2'_n} - \frac{1}{2} C^{0'}_{2'_n}\left[\left(C^{0'}_{2'_n}\right)^T C^{0'}_{2'_n} - I\right] \qquad (2\text{-}56)$$

where $C^{0'}_{2'_n}$ (ortho) is the result of performing the indicated operations on $C^{0'}_{2'_n}$. The former, $C^{0'}_{2'_n}$ (ortho), is restored in the same locations as $C^{0'}_{2'_n}$, and the same designation, $C^{0'}_{2'_n}$, is used

6. Compute Earth's Rate Signals

The angular velocity of the earth fixed (0') frame expressed in the computational (2') frame is computed,

$$\underline{\omega}^{2'}_{0,0'_n} = \left(C^{0'}_{2'_n}\right)^T \underline{\omega}^{0'}_{0,0'} = \{\sin L_n,\ -\cos L_n \sin\alpha_n,\ \cos L_n \cos\alpha_n\}^T \omega_{ie} \qquad (2\text{-}57)$$

where $\underline{\omega}^{0'}_{0,0'_n} = C^{0'}_{0_n}\,\underline{\omega}^{0}_{0,0'_n}$

since $C^{0'}_{0_n} = Z(\omega_{0,0'}\, t)$

and $\underline{\omega}^{0}_{0,0'} = \{0\ 0\ \omega^{0}_{0,0'}\}^{T}$

where $\omega^{0}_{0,0'} = \omega^{0'}_{0,0'} = \omega_{ie}$ is the constant magnitude of angular velocity of the earth in the inertial frame. ω_{ie} is entirely along the z axis of the inertial or earth fixed frames.

7. Compute Gyro Torquing Signals

Gyro torquing signals, $\underline{\omega}^{2'}_{0,2'_n}$ are computed viz:

$$\underline{\omega}^{2'}_{0,2'_n} = \underline{\omega}^{2'}_{0,0'_n} + \underline{\omega}^{2'}_{0',2'_n} \qquad (2\text{-}58)$$

where $\underline{\omega}^{2'}_{0,2'}$ is the total angular velocity of the LVWA gyros with inertial frame required to maintain the LVWA platform in the desired orientation (no gyro drift).

NOTE: (a) the values shown in (2-58) are those rates which will be applied continuously from t_n to t_{n+1}. If the angle to be torqued is required both sides of the equation are multiplied by Δt.

8. Compute Angular Velocity (for $\underline{\omega}x\underline{v}$)

The angular velocity for the Coriolis and centripetal terms is computed viz:

$$\underline{\omega}^{2'}_{0',2'_n} + 2\underline{\omega}^{2'}_{0,0'_n} = \underline{\omega}^{2'}_{0,2'_n} + \underline{\omega}^{2'}_{0,0'_n} \triangleq \underline{\omega}_n \qquad (2\text{-}59)$$

where $\underline{\omega}_n$ is introduced for compactness of notation.

NOTE: For the velocity update function, $\underline{\omega}_n$ is multipled by Δt.

9. Compute $(\underline{\omega}x\underline{v})\Delta t$ (for velocity update)

The Coriolis and centripetal portion of the velocity update is computed viz:

$$\left(\underline{\omega}_n x \underline{v}_n^{2'}\right)\Delta t = -\left(\Omega^{2'}_{0',2'_n} + 2\Omega^{2'}_{0,0'_n}\right)\underline{v}_n^{2'}\Delta t \qquad (2\text{-}60)$$

where $\Omega^{2'}_{0,0'}$ is the skew symmetric form of $\underline{\omega}^{2'}_{0,0'_n}$

and all other terms have been defined earlier.

10. Compute Altitude and Vertical Damping

The altitude and vertical damping terms are computed viz:

$$h_n = h_{n-1} + \left(v_{x_{n-1}}^{2'} + C_1\ \delta h_{n-1}\right)\Delta t$$

$$\delta h_n = h_{B_n} - h_n \qquad (2\text{-}61)$$

$$\hat{\delta a}_n = \hat{\delta a}_{n-1} + C_3\ \delta h_n \Delta t$$

where h_n is the system computed altitude at time t_n

C_1 is a vertical damping constant (sec^{-1})

h_{B_n} is the (barometric) reference altitude at t_n

δh_n is the difference between the barometric and system computed altitudes at time t_n.

C_3 is a vertical damping constant (sec^{-3}) (C_3 may be equal to zero).

$\hat{\delta a}_n$ is C_3 times the approximate integral of the computed altitude difference from t=0 to $t=t_n$.

11. Compute Gravity Components

The components of gravity in the computational frame $\underline{g}_n^{2'}$ are computed viz:

$$g_{x_n}^{2'} = -\left[c_{g_o} + c_{g_2}\left(c_{2',n}^{0'}\right)_{31}^2 + c_{g_4}\left(c_{2',n}^{0'}\right)_{31}^2\right]\left\{1 - \left[c_{h_1} - c_{h_4}\left(c_{2',n}^{0'}\right)_{31}^2\right]h_n + c_{h_2}\,h_n^2\right\}$$

$$g_{y_n}^{2'} = c_{h_o}\,h_n\left(c_{2',n}^{0'}\right)_{31}\left(c_{2',n}^{0'}\right)_{32} \qquad (2\text{-}62)$$

$$g_{z_n}^{2'} = -c_{h_o}\,h_n\left(c_{2',n}^{0'}\right)_{31}\left(c_{2',n}^{0'}\right)_{33}$$

where $g_{i_n}^{2'}$ for i=x,y,z, are the components of gravity in the computational (2') frame computed at t_n

and c_{g_o}, c_{g_2}, c_{g_4} are constants in the gravity equations with units of $(\text{feet/sec})^2$

c_{h_1}, c_{h_4} are constants in the gravity equations with units of (feet^{-1})

c_{h_2} is a constant in the gravity equations with unit of (feet^{-2})

c_{h_o} is a constant in the gravity equations with units of (sec^{-2})

$\left(c_{2',n}^{0'}\right)_{3j}$, where j = 1, 2, 3, are the elements of the third row of matrix $c_2^{0'}$ at time t_n

NOTE: (a) Before being applied as corrections in the velocity equation (2-52) the $g^{2'}_{i_n}$ are multiplied by Δt

(b) • For compatibility with the accuracy of the remainder of the "baseline" algorithm, the constants, C_{g_4}, C_{h_o}, C_{h_2}, C_{h_4}, should be set to zero. This would provide a gravity model of slightly greater accuracy than those employed by most of the systems surveyed.

• The form and constants employed in Equation (2-62) are those of the WGS72 Ellipsoidal Earth Gravity Model (same as "upgraded" algorithm).

• The value of these parameters are not under operator control.

• Effects of gravity model variations on navigation and attitude accuracy are not among the parameters to be included in this study.

12. Extract Position and Wander Angle

Position and wander angle are extracted from the direction cosine matrix, $C^{0'}_{2'_n}$, viz:

$$L_n = \tan^{-1}\left\{\frac{\left(C^{0'}_{2'n}\right)_{31}}{\left[\left(C^{0'}_{2'n}\right)^2_{11} + \left(C^{0'}_{2'n}\right)^2_{21}\right]^{1/2}}\right\} \qquad (2\text{-}63)$$

$$\lambda_n = \tan^{-1}\left[\frac{\left(C^{0'}_{2'n}\right)_{21}}{\left(C^{0'}_{2'n}\right)_{11}}\right] + \lambda_0 \qquad (2\text{-}64)$$

$$\alpha_n = \tan^{-1}\left[\frac{-\left(C^{0'}_{2'n}\right)_{32}}{\left(C^{0'}_{2'n}\right)_{33}}\right] \qquad (2\text{-}65)$$

where L_n is the geodetic latitude (positive north of equator) at time t_n

λ_n is the geodetic longitude (positive east of Greenwich) at time t_n.

λ_0 is the geodetic longtiude at t-0

α_n is the wander angle (positive to the east of north) at t_n

NOTE: If $\left(C^{0'}_{2'n}\right)_{11}$ and $\left(C^{0'}_{2'n}\right)_{21}$ are both zero than the system is exactly at the north or south pole and both λ_n and α_n are indeterminate therefore the previous values will be output as long as these conditions persist.

13. Form Wander Angle Matrix and LVN Velocities

Sine and cosine of wander angle are used to transform velocity from computational (2') frame to local vertical north pointing (LVN or 2) frame viz:

$$\underline{v}^2_n = C^2_{2'n}\,\underline{v}^{2'}_n \tag{2-66}$$

where $C^2_{2'_n} = X(\alpha_n) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_n & \sin\alpha_n \\ 0 & -\sin\alpha_n & \cos\alpha_n \end{bmatrix}$

and $\underline{v}^2_n$ is the earth relative velocity in the LVN frame at t_n.

2.4.3.1 Upgrading Baseline Navigation Algorithm

2.4.3.1.1 Introduction

The "baseline" navigation algorithm presented in the previous section may be considered as representative of moderate accuracy, local vertical, inertial navigators developed in the mid-to-latter 1960's.

The Inertial Measurement Unit (IMU) generally interfaced with a dedicated navigation computer, which was normally a small, fixed point, general purpose machine, typically with a 24-bit data word. Speed was moderate and operations like division were to be avoided if at all possible, while subroutines like square roots were not considered for the high speed loops. Navigation computation cycles were on the order of 2 to 20 Hz.

Considerable ingenuity was devoted to the development of "baseline" algorithms that primarily involved the arithmetic operations of addition and multiplication plus loads, stores, and shifts.

Further ingenuity was involved in optimum scaling of the variables; the use of binary computation cycles for the integrations obviated the necessity to perform multiplications in many cases or replaced the multiplication by a shift.

The net result was a compact, efficient, assembly language program, optimized - insofar as the constraints of time and money permitted - for the particular computer, IMU, and mission.

The local vertical wander azimuth navigation algorithms surveyed integrated linear acceleration once to obtain linear velocity (three components); converted linear velocity to angular velocity, and integrated the latter once to obtain angular position.

First order algorithms were used throughout. In integrating the $\dot{v}$ ($\dot{}$ denotes differentiation with respect to time) equation, it was assumed that the acceleration, $\underline{a}$, was constant over the navigation computation cycle, Δt; hence

$$\int_{t}^{t+\Delta t} \underline{a}(\tau)d\tau = \underline{a}(t)\ \Delta t \qquad (2\text{-}67)$$

This implied that $\dot{a}(t)$, $\ddot{a}(t)$, etc. were all equal to zero. Similarly, in integrating angular velocity, $\underline{\omega}$, it was assumed that $\underline{\omega}$ was constant over the navigation computation cycle, Δt, and further that $|\underline{\omega}| = (\underline{\omega}^T\underline{\omega})^{1/2}$, is sufficiently small that the $\omega_i\omega_j$ terms are negligible; hence the direction cosine matrix, C, containing the angular position variables is updated by

$$C(t + \Delta t) = C(t)\ [I + \Omega\ \Delta t] \qquad (2\text{-}68)$$

where I is the 3x3 identity matrix and Ω is the 3x3 skew symmetric matrix corresponding to $\underline{\omega}$. This implied that $\dot{\underline{\omega}}(t)$, $\ddot{\underline{\omega}}(t)$, etc. were all equal to zero. Compatible approximations in the conversion of linear to angular velocity, in the computation of gravity (the resultant of mass attraction and centripetal force), and in the trigonemetric (sine-cosine) and inverse trigonometric (arctangent) functions, were also part of the "baseline" navigation software.

The objective of the algorithm upgrade operation is the removal some of these assumptions or approximations on an as-needed basis to permit the airborne navigation computations to approach the accuracy of the PROFGEN outputs, which are considered to be perfect.

2.4.3.2 Approach

The "baseline" navigation equations were incorporated into the numerical simulator (NUMSIM) and a number of short simulations were performed. The errors in position, velocity and attitude relative to PROFGEN output were examined and the most significant were analyzed, using expanded printouts where necessary. When the offending routine was identified, its performance was analyzed and a more precise routine was devised, coded and tested. Then the procedure was repeated for the most significant remaining errors. The user is provided the options of "turning on" the upgraded algorithms by setting one or more flags to non zero values. This procedure is described in the User's Guide in Volume III.

2.4.4 Improved Accuracy Navigation Algorithm

The equations mechanized for the "ideal simulator LVWA Navigation Algorithm incorporate all the "upgrades" discussed in the previous section. In the main, the upgrades consist of a certain amount of interpolation and extrapolation of the navigation variables. The object of these operations is to correct the "phasing" of the scalars, vector, and matrix variables so that a higher order integration algorithm is effectively mechanized — but with little more computational load than is required for the simple "baseline" software.

The main addition to the algorithm is the repetition of the angular velocity computations — for the first pass, the interpolated velocities are used to generate angular velocities for the direction cosine matrix update, while the second pass uses the extrapolated velocities to generate the torquing signals and the Coriolis and centripetal corrections.

The routines are summarized in Table 2-3. The form of the difference equations used in the routines is listed as follows:

1. Updated Computational Frame Velocities

$$\underline{v}_n^{2'}(u) = \underline{v}(d)_{n-1}^{2'} + \underline{\Delta v}_n^{2'} + \left[\underline{g}_{n-1/2}^{2'} - \left(\Omega_{0',2'_{n-1/2}}^{2'} + 2\Omega_{0,0'_{n-1/2}}^{2'}\right)\underline{v}_{n-1/2}^{2'}\right]\Delta t \qquad (2\text{-}69)$$

Note: The variables with subscript (n-1/2) are the extrapolated values computed at t_{n-1} i.e., (n - 1 + 1/2).

2. Add Vertical Damping Terms

$$v_{x_n}^{2'}(d) = v_{x_n}^{2'}(u) + \left(C_2\delta h_{n-1/2} + \hat{\delta a}_{n-1/2}\ \underline{v}_{n-1/2}^{2'}\right)\Delta t \qquad (2\text{-}70)$$

Note: The (d) and (u) are not employed in the subsequent equations since only v(d) is saved.

TABLE 2-3

Routines for

Upgraded LVWA Navigation Algorithm with Vertical Damping

Sequence No.*	Function	Comments
1.	Update computational frame velocities	from (n-1) to n, store old velocities
2.	Add vertical Damping terms	
2a.	Interpolate and extrapolate velocities	to (n-1/2) and (n+1/2)
3.	Compute angular velocities, $\underline{\rho}$, about level axes, for d.c.m. update, using interpolated- updated velocities and extrapolated α, L, h.	(at (n-1/2)
4.	Update direction cosine matrix	to n, using $\underline{\rho}$ from 3 and 2nd order algorithm
4a.	Extrapolate direction cosine matrix	to (n+1/2)
5.	Orthonomalize direction cosine matrix	
10.	Compute altitude and vertical damping	extrapolate h, δh, store as old value
11.	Compute gravity components	using extrapolated L, h
12.	Extract position and wander angle	store old α
13.	Form wander angle matrix and LVN velocities	extrapolate wander angle sin - cos
3a.	Compute angular velocities about level axes	extrapolated to (n+1/2), for torquing and ($\underline{\omega}$ x $\underline{v}$) using extrapolated $\underline{V}$, $\underline{\alpha}$, L, h
6.	Compute earth's rate	using extrapolated d.c.m., $\underline{V}$, $\underline{\rho}$
7.	Compute gyro torquing	
8.	Compute angular velocity for ($\underline{\omega}$ x $\underline{V}$)	
9.	Compute ($\underline{\omega}$ x $\underline{V}$) Δt for velocity update	

*Sequence numbers correspond to those used in the "baseline" algorithm (see Table 2-1). Additional or repeated steps are indicated by a suffix "a".

2a. Interpolate and Extrapolate Velocities [to (n-1/2) and (n+1/2)]

$$\underline{v}^{2'}_{n-1/2} = \frac{1}{2}\left(\underline{v}^{2'}_{n} + \underline{v}^{2'}_{n-1}\right) \tag{2-71}$$

$$\underline{v}^{2'}_{n+1/2} = \frac{1}{2}\left(3\underline{v}^{2'}_{n} - \underline{v}^{2'}_{n-1}\right)$$

3. Compute Angular Velocities about Level Axes

Transform level components of velocity to LVN frame

$$\begin{aligned} v^{2}_{y_{n-1/2}} &= v^{2'}_{y_{n-1/2}} \cos\alpha_{n-1/2} + v^{2}_{z_{n-1/2}} \sin\alpha_{n-1/2} \\ v^{2}_{z_{n-1/2}} &= -v^{2'}_{y_{n-1/2}} \sin\alpha_{n-1/2} + v^{2'}_{z_{n-1/2}} \cos\alpha_{n-1/2} \end{aligned} \tag{2-72}$$

Compute E and N components of angular velocity

$$\omega_{E_{n-1/2}} = \frac{-v^{2}_{z_{n-1/2}}}{h_{n-1/2} + \left\{ \dfrac{r_o(1-\varepsilon)^2}{\left[1-\varepsilon^2\left(C^{o'}_{2'_{n-1/2}}\right)^2_{31}\right]^{1/2}} \right\}} \tag{2-73}$$

$$\omega_{N_{n-1/2}} = \frac{-v^{2}_{z_{n-1/2}}}{h_{n-1/2} + \left\{ \dfrac{r_o}{\left[1-\varepsilon^2\left(C^{o'}_{2'_{n-1/2}}\right)^2_{31}\right]^{1/2}} \right\}}$$

Transform angular velocity to computational frame

$$\begin{aligned} \rho_{x_{n-1/2}} &= 0 \\ \rho_{y_{n-1/2}} &= \omega_{E_{n-1/2}} \cos\alpha_{n-1/2} - \omega_{N_{n-1/2}} \sin\alpha_{n-1/2} \\ \rho_{z_{n-1/2}} &= \omega_{E_{n-1/2}} \sin\alpha_{n-1/2} + \omega_{N_{n-1/2}} \cos\alpha_{n-1/2} \end{aligned} \tag{2-74}$$

where $i^{2}_{n-1/2}$, and i = y, z, are the east and north components of velocity at time $t_{n-1/2}$.

$\omega_{E_{n-1/2}}$ $\omega_{N_{n-1/2}}$ are the y and z components of $\underline{\omega}^{2}_{0',2}{}_{n-1/2}$ i.e., the level components of the angular rate of the vehicle with respect to the earth fixed frame in the LVN frame at $t_{n-1/2}$, the mid-point of the nth computation cycle.

$\rho_{x_{n-1/2}}$ $\rho_{y_{n-1/2}}$ $\rho_{z_{n-1/2}}$ are the x, y, and z components of $\omega^{2'}_{0',2'}{}_{n-1/2}$ i.e., the components of the angular rate of the vehicle with respect to the earth-fixed frame expressed in the computational frame at $t_{n-1/2}$.

$\left(C^{0'}_{2'}{}_{n-1/2}\right)^{2}_{31}$ is the square of the first element of the third row of the transformation from the computational frame to the earth fixed frame (the direction cosine matrix) at $t_{n-1/2}$. It is equal to $\sin^{2}L_{n-1/2}$.

4. Update Direction Cosine Matrix

The direction cosine matrix, $C^{0'}_{2'}{}_{n-1}$ is updated (to the second order) using the $\rho_{i_{n-1/2}}$ computed above viz:

$$C^{0'}_{2'n} = C^{0'}_{2'}{}_{n-1} + C^{0'}_{2'}{}_{n-1}\left[\Omega^{2'}_{0',2'}{}_{n-1/2}\,\Delta t + \left(\Omega^{2'}_{0',2'}{}_{n-1/2}\right)^{2}\frac{\Delta t^{2}}{2}\right] \qquad (2\text{-}75)$$

where $\Omega^{2'}_{0',2'_{n-1/2}}$ is the skew symmetric form of $\underline{\rho}_{n-1/2}$ or $\underline{\omega}^{2'}_{0',2'_{n-1/2}}$ and $\left(\Omega^{2'}_{0,2'_{n-1/2}}\right)^2$ is the square of the above matrix.

Note: The nature of the interpolated $\underline{\rho}_{n-1/2}$ is such that some portions of a third and higher order update are incorporated, while the form is only second order (see Appendix D).

4a. Extrapolate Updated Direction Cosine Matrix

$$C^{0'}_{2'_{n+1/2}} = C^{0'}_{2'_n} + \frac{1}{2}\left[C^{0'}_{2'_{n-1}} \Omega^{2'}_{0',2'_{n-1/2}} \Delta t + C^{0'}_{2'_{n-1}} \left(\Omega^{2'}_{0',2'_{n-1/2}}\right)^2 \frac{\Delta t^2}{2}\right] \quad (2\text{-}76)$$

where $C^{0'}_{2'_{n+1/2}}$ is $C^{0'}_{2'_n}$ extrapolated to the midpoint of the next computation cycle

and $\left[\quad\right]$ is the update matrix added to $C^{0'}_{2'_{n-1}}$ to update it to $C^{0'}_{2'_n}$ (see Equation 2-75)

5. Orthonomalize Direction Cosine Matrix

This operation is identical to that performed in the "baseline" case. Equation (2-56) is repeated below for completeness.

$$C^{0'}_{2'_n}(\text{ortho}) = C^{0'}_{2'_n} - \frac{1}{2} C^{0'}_{2'_n}\left[\left(C^{0'}_{2'_n}\right)^T C^{0'}_{2'_n} - I\right] \quad (2\text{-}56)$$

10. Compute Altitude and Vertical Damping

The altitude and vertical damping terms are computed viz:

$$h_n = h_{n-1} + \left(v^{2'}_{x_{n-1/2}} + C_1 \, \delta h_{n-1/2} \right) \Delta t$$

$$h_{n+1/2} = h_n + \frac{1}{2} v^{2'}_{x_n} \Delta t$$

$$h_{B_{n+1/2}} + \frac{1}{2} \left(3h_{B_n} - h_{B_{n-1}} \right) \qquad (2\text{-}77)$$

$$\delta h_{n+1/2} = h_{B_{n+1/2}} - h_{n+1/2}$$

Store $h_{B_{n+1/2}}$ as $h_{B_{n-1}}$

$$\hat{\delta a}_{n+1/2} = \hat{\delta a}_{n-1/2} + C_3 \delta h_{n+1/2} \Delta t$$

11. Compute Components of Gravity

The components of gravity in the computational (LVWA) frame, at $t_{n+1/2}$, are computed for the WGS 72 Ellipsoid viz:

$$g^{2'}_{x_{n+1/2}} = \left[C_{g_0} + C_{g_2} \left(C^{0'}_{2'_{n+1/2}} \right)^2_{31} + C_{g_0} \left(C^{0'}_{2'_{n+1/2}} \right)^2_{31} \right] \left\{ 1 - \left[C_{h_1} - C_{h_4} \left(C^{0'}_{2'_{n+1/2}} \right)^2_{31} \right] h_{n+1/2} + C_{h_2} h^2_{n+1/2} \right\}$$

$$g^{2'}_{y_{n+1/2}} = C_{h_0} \, h_{n+1/2} \left(C^{0'}_{2'_{n+1/2}} \right)_{31} \left(C^{0'}_{2'_{n+1/2}} \right)_{32} \qquad (2\text{-}78)$$

$$g_{z_{n+1/2}} = -C_{h_0} \, h_{n+1/2} \left(C^{0'}_{2'_{n+1/2}} \right)_{31} \left(C^{0'}_{2'_{n+1/2}} \right)_{33}$$

where $g^{2'}_{i_{n+1/2}}$, for i = x, y, z, are the components of gravity in computational (2') frame at $t_{n+1/2}$.

$\left(C^{0'}_{2'}{}_{n+1/2}\right)_{3j}$ for j = 1, 2, 3, are the elements of the third row of the transformation from the computational frame to the earth-fixed frame, extrapolated to $t_{n+1/2}$.

$h_{n+1/2}$ is the altitude extrapolated to $t_{n+1/2}$.

Note: All the constants in Equation (2-78) the same as in Equation (2-62).

12. Extract Position and Wander Angle

This operation is identical to that performed in the "baseline" case. Equations (2-63, 2-64, 2-65) are repeated below for completeness:

$$L_n = \tan^{-1}\left\{\frac{\left(C^{0'}_{2'n}\right)_{31}}{\left[\left(C^{0'}_{2'n}\right)^2_{11} + \left(C^{0'}_{2'n}\right)^2_{21}\right]^{1/2}}\right\} \qquad (2\text{-}63)$$

$$\lambda_n = \tan^{-1}\left[\frac{\left(C^{0'}_{2'n}\right)_{21}}{\left(C^{0'}_{2'n}\right)_{11}}\right] + \lambda_0 \qquad (2\text{-}64)$$

$$\alpha_n = \tan^{-1}\left[\frac{-\left(C^{0'}_{2'n}\right)_{32}}{\left(C^{0'}_{2'n}\right)_{33}}\right] \qquad (2\text{-}65)$$

Note: Before α_n is computed, the previous value of α is stored as α_0.

13. Form Wander Angle Matrix and LVN Velocities

The fundamental operation is identical to that performed in the "baseline" case. Equation (2-66) is repeated for completeness.

$$\underline{v}_n^2 = C_{2'}^2 \; \underline{v}_n^{2'} \qquad (2\text{-}66)$$

In addition, the sine and cosine of the wander angle are extrapolated to $t_{n+1/2}$.

$$\delta\alpha = 1/2\ (\alpha_n - \alpha_0)$$

$$\sin \alpha_{n+1/2} = \sin \alpha_n + \delta\alpha \cos \alpha_n \qquad (2\text{-}79)$$

$$\cos \alpha_{n+1/2} = \cos \alpha_n - \delta\alpha \sin \alpha_n$$

where $\sin \alpha_{n+1/2}$ $\cos \alpha_{n+1/2}$ are the sine and cosine respectively of the wander angle, extrapolated forward to mid-computation cycle.

3a. Compute Angular Velocities About Level Axes

The angular velocities about the level axes of the computational frame are computed using the same formulae as in Equation (2-73), but with all the variables extrapolated to $t_{n+1/2}$.

Transform level components of velocity to LVN frame

$$v^2_{y_{n+1/2}} = v^{2'}_{y_{n+1/2}} \cos \alpha_{n+1/2} + v^{2'}_{z_{n+1/2}} \sin \alpha_{n+1/2}$$

$$v^2_{z_{n+1/2}} = v^{2'}_{y_{n+1/2}} \sin \alpha_{n+1/2} + v^{2'}_{z_{n+1/2}} \cos \alpha_{n+1/2} \qquad (2\text{-}80)$$

Compute E and N components of angular velocity

$$\omega_{E_{n+1/2}} = \frac{-v_{z_{n+1/2}}^2}{h_{n+1/2}\left\{\frac{r_0\,(1-\varepsilon^2)}{\left[1-\varepsilon^2\left(C_{2',n+1/2}^{0'}\right)_{31}^2\right]^{3/2}}\right\}}$$

$$\omega_{N_{n+1/2}} = \frac{v_{Y_{n+1/2}}^2}{h_{n+1/2}+\left\{\frac{r_0}{\left[1-\varepsilon^2\left(C_{2',n+1/2}^{0'}\right)_{31}^2\right]^{1/2}}\right\}} \qquad (2\text{-}81)$$

Transform angular velocity to computational frame

$$\rho_{x_{n+1/2}} \triangleq 0$$

$$\rho_{y_{n+1/2}} = \omega_{E_{n+1/2}} \cos\alpha_{n+1/2} - \omega_{N_{n+1/2}} \sin\alpha_{n+1/2} \qquad (2\text{-}82)$$

$$\rho_{z_{n+1/2}} = \omega_{E_{n+1/2}} \sin\alpha_{n+1/2} + \omega_{N_{n+1/2}} \cos\alpha_{n+1/2}$$

Note: (a) The variable definitions accompanying Equation (2-73) apply here - only the time has been advanced from $t_{n-1/2}$ to $t_{n+1/2}$.

(b) The values of $\sin\alpha_{n+1/2}$ and $\cos\alpha_{n+1/2}$ are stored in locations for the "extrapolated sin-cos of α", and will be used as $\sin\alpha_{n-1/2}$ and $\cos\alpha_{n-1/2}$ when Equation 2-73 is used on the next computation cycle.

(c) A similar comment to (b) above, applies to $\left(C_{2',n+1/2}^{0'}\right)$.

6. Compute Earth's Rate Signals

The earth's rate signals are computed in the same manner as for the "baseline" case, (Equation (2-57)), except that the extrapolated direction cosine matrix, $C^{0'}_{2',n+1/2}$, is employed.

$$\omega^{2'}_{0,0'_{n+1/2}} = \left(C^{0'}_{2',n+1/2}\right)^T \underline{\omega}^{0'}_{0,0'} \qquad (2\text{-}83)$$

$$= \left\{ \sin L_{n+1/2}, -\cos L_{n+1/2} \sin\alpha_{n+1/2}, \cos L_{n+1/2} \cos\alpha_{n+1/2} \right\} \omega_{ie}$$

7. Compute Gyro Torquing Signals

Gyro torquing signals, $\underline{\omega}^{2'}_{0,2',n+1/2}$, are computed in the identical manner to that employed in the "baseline" case, Equation (2-58) except that the variables are extrapolated to $t_{n+1/2}$.

$$\underline{\omega}^{2'}_{0,2'_{n+1/2}} = \underline{\omega}^{2'}_{0,0'_{n+1/2}} + \underline{\omega}^{2'}_{0',2'_{n+1/2}} \qquad (2\text{-}84)$$

where $\underline{\omega}^{2'}_{0',2'_{n+1/2}}$ is $\underline{\rho}_{n+1/2}$ as computed in Equation (2-81).

8. Compute Angular Velocity (for $\underline{\omega} \times \underline{v}$)

Angular velocity for the Coriolis and centripetal terms, $\underline{\omega}_{n+1/2}$ is computed as in Equation (2-59) for the "baseline" case, except that the extrapolated variables are used:

$$\underline{\omega}_{n+1/2} = \underline{\omega}^{2'}_{0,2'_{n+1/2}} + \underline{\omega}^{2'}_{0,0'_{n+1/2}} \qquad (2\text{-}85)$$

9. Compute $(\underline{\omega} \times \underline{v})\ \Delta t$ (for velocity update)

The Coriolis and centripetal portion of the velocity update is computed in the same manner as in Equation (2-60) for the "baseline" case, except that the extrapolated values (to $t_{n+1/2}$) are employed.

$$-\left(\underline{\omega}_{n+1/2} \times \underline{v}^{2'}_{n+1/2}\right) \Delta t = -\left(\Omega^{2'}_{0',2'_{n+1/2}} + 2\Omega^{2'}_{0,0'_{n+1/2}}\right) \underline{v}^{2'}_{n+1/2}\ \Delta t \qquad (2\text{-}86)$$

General Note: The extrapolated variables computed by Equation (2-77) through (2-86), at t_n, become the interpolated variables used in Equations (2-69) through (2-73), at t_{n+1}.

2.5 Attitude Computations

Attitude computations for the three IMU types, all employing a local vertical wander azimuth (LVWA) computational frame, differ only in the manner of constructing the transformation from the body frame to the LVWA computational frame C_b^c or $C_5^{2'}$.

2.5.1 Local Vertical Wander Azimuth IMU

A four gimbal, local vertical IMU is considered since attitude may be read out directly in the three gimbal case. In this case, $C_5^{2'}$ is the LVWA gimbal angle matrix. The generation of this matrix is described in detail in Appendix A, Section A.4.7 of this volume.

2.5.2 Space Stablized IMU

All gimbaled, SS IMU's have four gimbals. The SS gimbal angle matrix or transformation from the body frame to the space stable platform frame, C_b^p or C_5^{10}, is described in detail in Appendix A, Section A.4.6 of this volume. However, this is only the first step in generating $C_5^{2'}$ for a space stable IMU with a LVWA computational frame. The total transformation is given by:

$$C_b^c = C_e^c \, C_i^e \, C_p^i \, C_b^p$$

or

$$C_5^{2'} = C_{0'}^{2'} \, C_0^{0'} \, C_{10}^0 \, C_5^{10}$$

where C_{10}^0 or C_p^i is the space stable alignment matrix (a constant for this study) - see Appendix A, Section A.4.8 of this volume.

$C_0^{0'}$ or C_i^e is the single axis rotation representing the inertial to earth fixed transformation - see same section as above.

$C_{0'}^{2'}$ or C_e^c is the transpose of the LVWA "direction cosine matrix - see Appendix A, Section A.4.5 of this volume.

2.5.3 Strapdown IMU

The body-to-computational frame transformation, C_b^c, is an integral part of the (high speed) strapdown computations. Its generation is described in detail in Section 2.3.3 of this volume.

2.5.4 Attitude Extraction

This operation is common to all IMU types with a LVWA computational frame, and is described in detail in Sections A.4.4 and A.4.9 of this volume.

It should be noted that roll and yaw are indeterminate when pitch is $\pm$ 90°.

2.5.5 Attitude Predictor-Filter

The attitude predictor filter accepts the computed attitude angles (roll, pitch and heading) as inputs and outputs filtered/predicted values of the attitude angles their angular rates and angular accelerations. It employs precomputed gains, selected on the basis of the observation residuals (the difference between the computed and estimated attitude angles). If the residual is less than a preassigned maximum, the gains are incremented towards the steady state values; conversely, if the residual equals or exceeds the maximum, gains are decremented toward the initial values, where complete reliance is placed on the computed attitude angle. The procedure is repeated for each attitude angle.

A block diagram of the filter is presented below (for a single attitude angle):

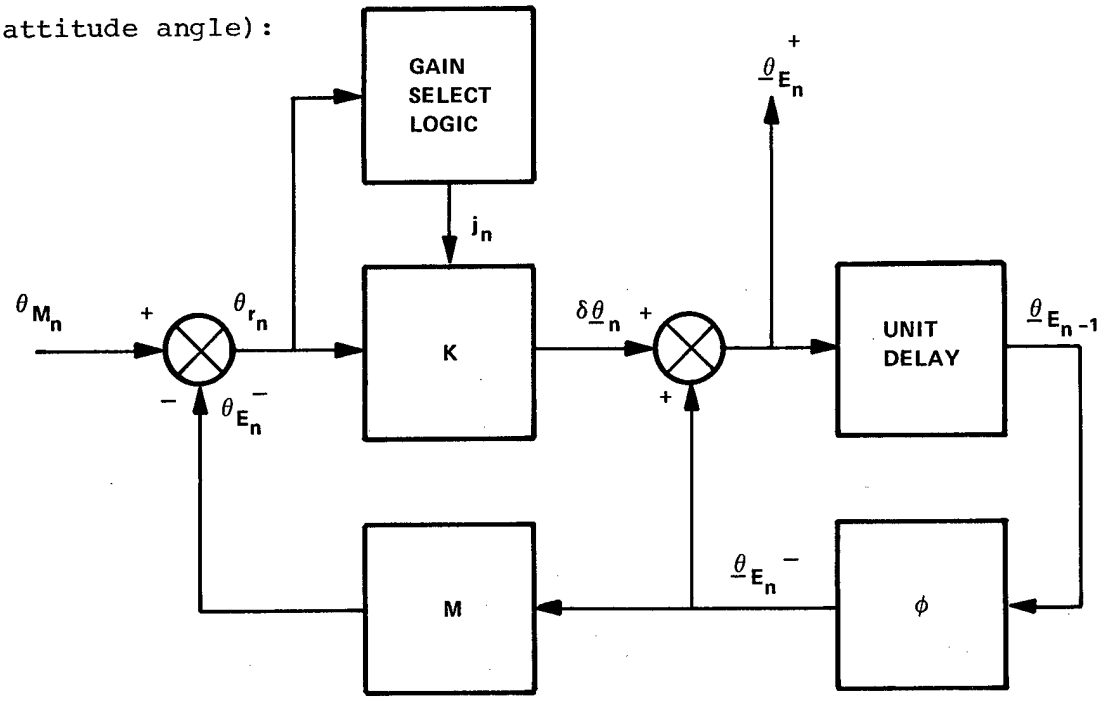


Figure 2-11. Attitude Predictor-Filter Block Diagram (Single Axis)

where θ_{M_n} is the computed (measured) attitude angle at t_n.

$\theta_{E_n}^-$ is the estimated attitude angle at t_n, prior to the update.

$\theta_{r_n} = \theta_{M_n} - \theta_{E_n}^-$ is the observation residual at t_n.

K is the 3x10 gain matrix.

j_n is the column of the gain matrix from the gain select logic, where $j_n = j_{n-1} \pm 1$ unless K = 10.

$M = \{1\ 0\ 0\}$ is the measurement matrix

Φ is the state transition matrix

where $\Phi \triangleq \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$

$\underline{\delta\theta}_n = \underline{K}_{j_n} \Delta\theta_n$ is the update to the attitude state, $\{\delta\theta_{E_n}\ \dot{\delta\theta}_{E_n}\ \ddot{\delta\theta}_{E_n}\}^T$.

$\underline{\bar{\theta}}_{E_n} = \{\bar{\theta}_{E_n}\ \dot{\bar{\theta}}_{E_n}\ \ddot{\bar{\theta}}_{E_n}\}^T$ is the estimated attitude state at t_n, prior to the update,

$\underline{\theta}^+_{E_n} = \{\bar{\theta}_{E_n} + \delta\theta_{E_n}\ \dot{\bar{\theta}}_{E_n} + \dot{\delta\theta}_{E_n}\ \ddot{\bar{\theta}}_{E_n} + \ddot{\delta\theta}_{E_n}\}^T$ is the estimated attitude state at t_n, after the upate.

Δt is the attitude computation cycle.

Three passes through the filter are required to update all three attitude angles and their derivatives, which are then stored in a 3x3 attitude state matrix.

It should be noted that the particular gain matrix, K, was that considered applicable for the SPN/GEANS, and may only be applied when attitude is computed at 32 Hz. Generally, the gains are obtained from a two step simulation program. In the first step the vehicle dynamics, the estimated initial errors in angular position, velocity, and acceleration, the actual and estimated encoder noises, the iteration time, the ratio of the iteration to the desired truncation time, the iterations from which gains are to be stored, and several other control parameters are input. The run is performed, including a statistical analysis of the errors, using a recursive, exponentially age weighted, least squares filter. In the second step, the run is repeated using the precomputed gains (i.e. the same filter configuration as previously described) and the instantaneous and statistical errors are compared. A number of repetitions of the process, with judicious variation of parameters should lead to a best gain matrix,

for the vehicle dynamics, encoder characteristics and iteration time. This program does not currently exist at CSDL.

The filter equations are mechanized in the following manner (for a single axis):

1) Extrapolate attitude state

$$\begin{bmatrix} \theta^-_{E_n} \\ \dot{\theta}^-_{E_n} \\ \ddot{\theta}^-_{E_n} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{E_{n-1}} \\ \dot{\theta}_{E_{n-1}} \\ \ddot{\theta}_{E_{n-1}} \end{bmatrix}$$

Note that $\theta_{E_{n-1}}$ is $\theta^+_{E_n}$ one cycle (Δt) later

2) Form observation residual

$$\theta_{r_n} = \theta_{M_n} - \theta^-_{E_n}$$

3) Select gain column

See Flowchart (Figure 2-12)

4) Update attitude state

$$\begin{bmatrix} \theta^+_{E_n} \\ \dot{\theta}^+_{E_n} \\ \ddot{\theta}^+_{E_n} \end{bmatrix} = \begin{bmatrix} \theta^-_{E_n} \\ \dot{\theta}^-_{E_n} \\ \ddot{\theta}^-_{E_n} \end{bmatrix} + \begin{bmatrix} K_{1_j} \\ K_{2_j} \\ K_{3_j} \end{bmatrix} \theta_{r_n}$$

The above are the filtered values at t_n. In essence the above procedure is repeated twice more.

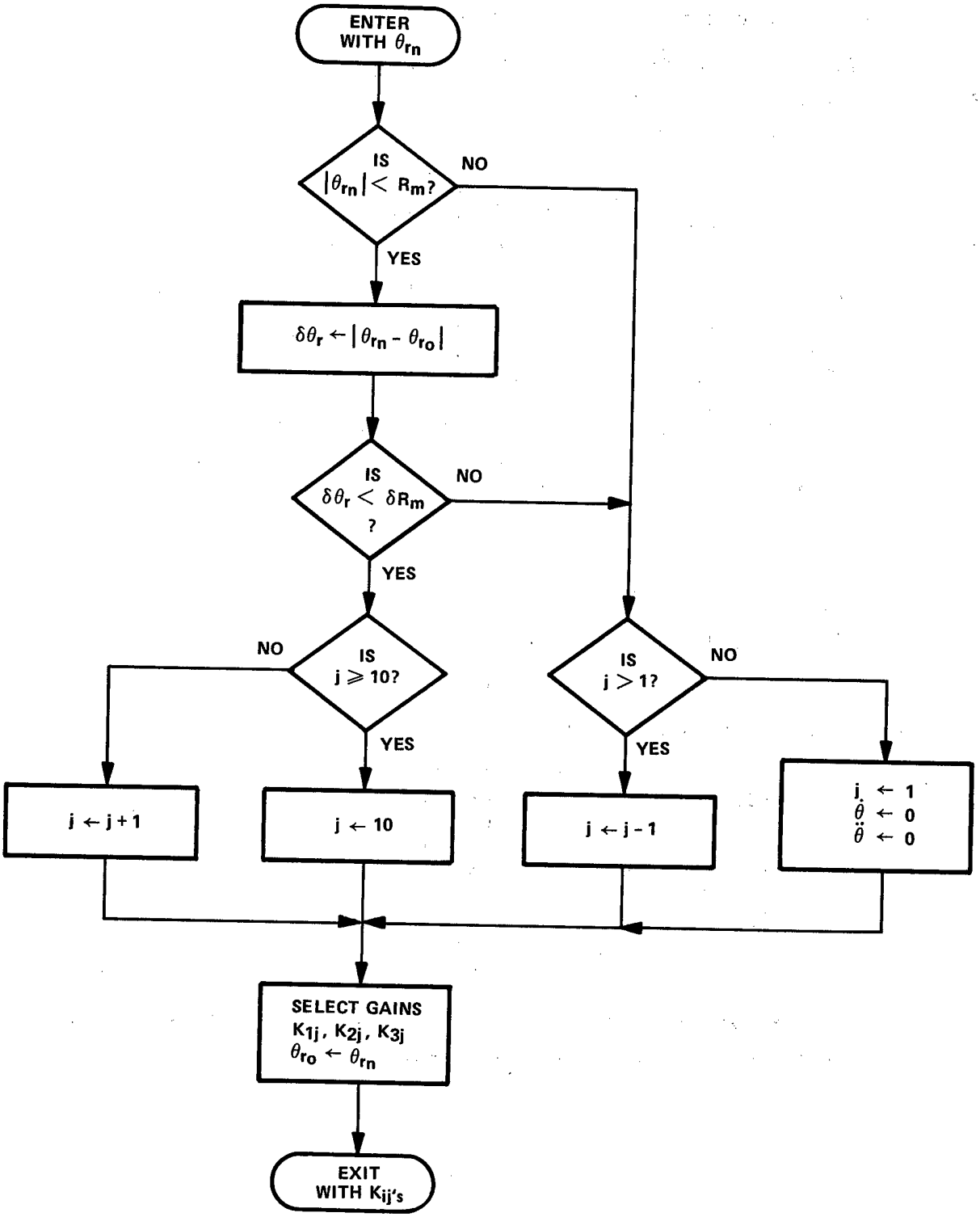


Figure 2-12. Attitude Filter Gain Select Logic for a Single Axis

SECTION 3

SUMMARY OF INS NAVIGATION COMPUTATIONS

3.1 Introduction

This Section contains a general functional description of the INS navigation software mechanizations, based on the survey results presented in Section 1.

The following INS configurations are included in this description:

1. INS with a local level stabilized platform
2. INS with a space stabilized platform
3. Strapdown INS

The computational frames included in this description, are:

1. Local level, geodetic, wander azimuth
2. Space stabilized

The resultant six types of computations are categorized in Figure 3-1. The figure numbers shown in the tabulation refer to the block diagrams of the navigation computations, presented in this Section.

The proposed standardized symbology of frames and transformations, presented in Section 6, is used in the overview described in this section.

3.2 Summary and Conclusions

The general functional description of INS software mechanization is used to illustrate the commonality and differences in navigation computations for three types of INS configurations (a local level, a space stabilized and a strapdown INS). Both the

INS CONFIGURATION → / COMPUTATIONAL FRAME ↓	LOCAL LEVEL STABILIZED PLATFORM	SPACE STABILIZED PLATFORM	STRAPDOWN
LOCAL LEVEL	FIG. 3-3, 3-4	FIG. 3-3, 3-5	FIG. 3-3, 3-6
SPACE STABILIZED	FIG. 3-7, 3-8	FIG. 3-7, 3-9	FIG. 3-7, 3-10

Figure 3-1. Categorization of the INS Computations

local-level wander-azimuth and the space-stabilized computational frames are included in this description.

The commonality of navigation computations of position and velocity in both computational frames starts with the velocity increments sensed by the accelerometers and resolved along the computational frame axes and ends with the position and velocity, computed in the local geographic frame.

The differences in navigation computations in both computational frames for the three types of INS configurations are represented by the following computations:

1. Transformation of the velocity increments sensed by the accelerometers from the accelerometers axes frame to the computational frame. The form of the transformation matrix is a function of the INS type.
2. Computation of the gyro torquing signals is required for the INS with local-level stabilized platform.

The attitude computations, irrespective of the computational frame used in the navigation computations, require determination of the transformation from the vehicle body axes frame to the local-geodetic, north-pointing frame.

For the INS with a gimballed platform, the gimbal angles are used to compute the transformation from the vehicle body frame to the platform frame. The transformation from the platform frame to the local-level, geodetic, north-pointing frame is determined using the computed navigation parameters. For the strapdown INS, the transformation from the body frame to the local-level, geodetic, north-pointing frame is computed using the angular velocities sensed by the gyros and the computed navigation parameters.

Selection of the computational frame used in the navigation computations determines the format of navigation parameters computations required for the attitude computations.

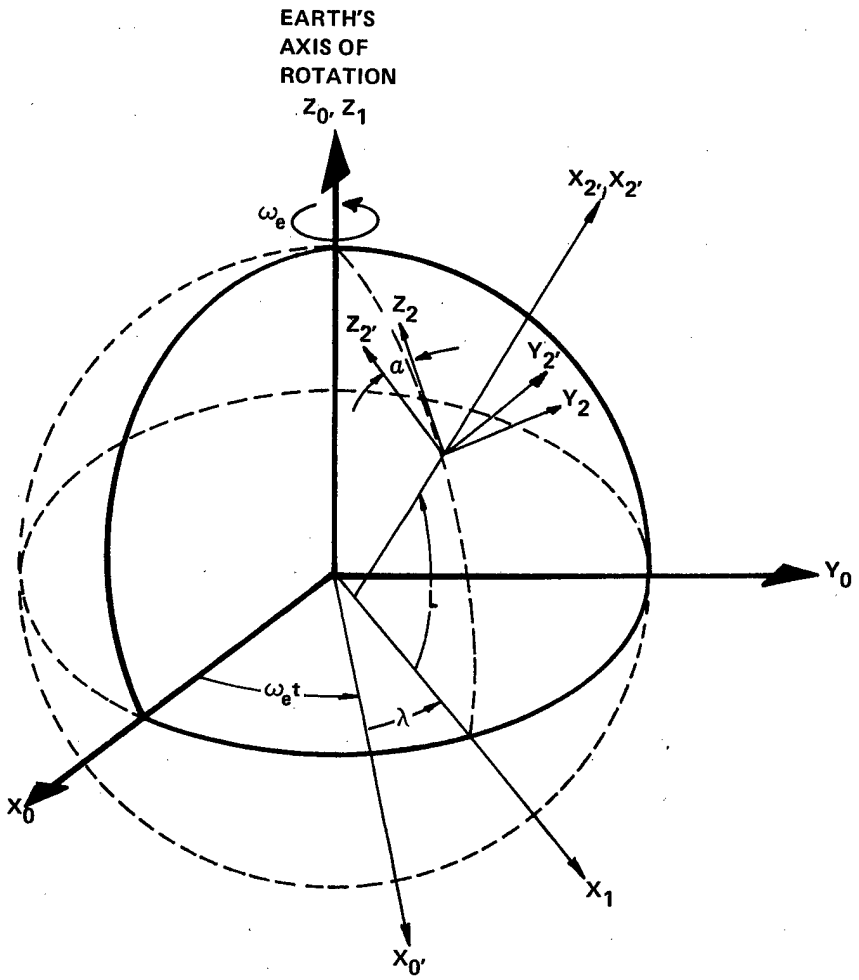
3.3 Frame Definition

Definition of the coordinate frames employed in this section presentation is shown in Figure 3-2. The frames definition is based on the proposed standardized symbology presented in Section 6.

3.4 Position and Velocity Computations in Local-Level Frame

Figure 3-3 presents a block diagram of the navigation computations of position and velocity in the local-level, geodetic, wander-azimuth computational frame. This part of the computations is common for the local level INS, the space stabilized INS and for the strapdown INS.

The velocity increments ΔV, sensed by the accelerometers and resolved along the axes Y_2, Z_2, are combined with the V/R terms and with the Coriolis correction terms. The computed horizontal velocity is resolved through the wander angle, α, into the east and the north components. Computation of the vertical velocity component requires the magnitude of the local gravity. In addition, the barometric altitude, h_B, is used in a third-order damping loop to bound the vertical channel computation errors. The computed horizontal velocity is used in the computation of the angular rate $\omega_{0,2'}$ of the computational frame, 2', with respect to the



FRAMES

$X_0 Y_0 Z_0$ = SPACE STABLIZED REFERENCE
$X_{0'} Y_{0'} Z_{0'}$ = EARTH FIXED REFERENCE, AT t = 0, 0' FRAME COINCIDES WITH 0 FRAME
$X_1 Y_1 Z_1$ = EARTH FIXED, FOR LONGITUDE $\lambda = 0$, 1 FRAME COINCIDES WITH 0' FRAME
X_2, Y_2, Z_2 = LOCAL GEODETIC UP, EAST, NORTH
$X_{2'}, Y_{2'}, Z_{2'}$ = LOCAL GEODETIC, WANDER AZIMUTH, WANDER ANGLE = α
X_5, Y_5, Z_5 = AIRCRAFT BODY X_5 UP, Y_5 ALONG RIGHT WING, Z_5 ALONG FUSELAGE (FORWARD)
FOR ATTITUDE ANGLES AND WANDER ANGLE = 0, THE 5 FRAME COINCIDES WITH THE 2 FRAME

Figure 3-2. Definition of Axes Frames

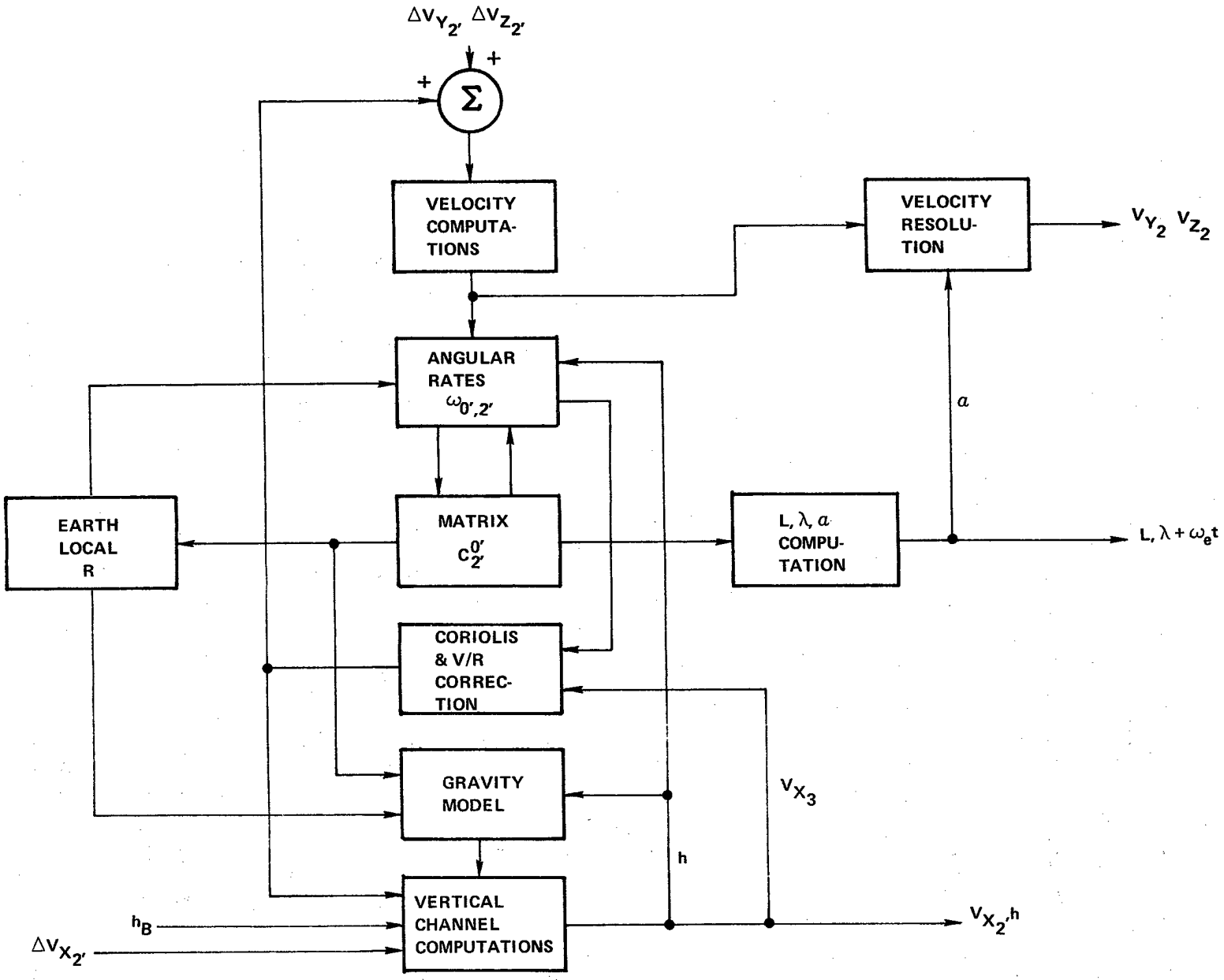


Figure 3-3. Commonality of Navigation Computations in a Local Geodetic, Wander Azimuth Frame - General Block Diagram

earth fixed frame, $0'$. The $\underline{\omega}_{0',2'}$ is used to update the direction cosine matrix of transformation, $C_{2'}^{0'}$, between the frames 2' and and 0'. The updating is based on solution of the differential equation $\dot{C}_{2'}^{0'} = C_{2'}^{0'} \omega_{0',2'}$. The latitude, L, longitude, $\lambda + \omega_{ie}$, and the wander angle, α, are computed from the $C_{2'}^{0'}$ matrix terms.

Figure 3-4 presents additional computations in the local level frame required for an INS with a local level platform stabilization. The gyro torquing signals are computed using the $C_{2'}^{0'}$ and the $\omega_{0',2'}$ terms.

Figure 3-5 presents additional computations in the local level frame required for an INS with a space stabilized platform. The velocity increments $\underline{\Delta V}^{10}$ sensed by the accelerometers mounted on the space stabilized platform are transformed from the platform axes frame, 10, to the computational frame, 2', using the updated directional cosines matrix $C_{10}^{2'} = C_2^{2'} C_1^2 C_0^1 C_0^{0'} C_{10}^0$.

Figure 3-6 presents additional computations in the local level frame required for a strapdown INS. The velocity increments $\underline{\Delta V}_5$ sensed by the accelerometers mounted on the aircraft body are transformed from the body frame, 5, to the computational frame, 2' using the direction cosines transformation matrix, $C_5^{2'}$. The matrix $C_5^{2'}$ is computed from the quaternion, g. The quaternion, g, is computed and updated using the angluar velocity $\underline{\omega}_{0,5}^{5}$ sensed by the gyros and the angular velocity $\underline{\omega}_{0,2'}^{5}$ of the computational frame, 2', with respect to the inertial frame, 0, computed in the 2' frame and transformed to the 5 frame.

3.5 Position and Velocity Computations in Space-Stabilized Frame

Figure 3-7 presents a block diagram of the navigation computations in the space-stabilized computational frame. This part of the computations is common for the local level INS, the space-stabilized INS and the strapdown INS.

The velocity increments, $\underline{\Delta V}^0$, sensed by the accelerometers and resolved along the space-stabilized axes frame, 0, and combined with the gravity contributions, $\underline{G}\Delta T$ and with the barometric damping term, $K_1 \Delta h$. The computed inertial velocity, $\underline{V}^0$, is used

- INS WITH A LOCAL LEVEL STABILIZED PLATFORM

Figure 3-4

- INS WITH A SPACE STABILIZED PLATFORM

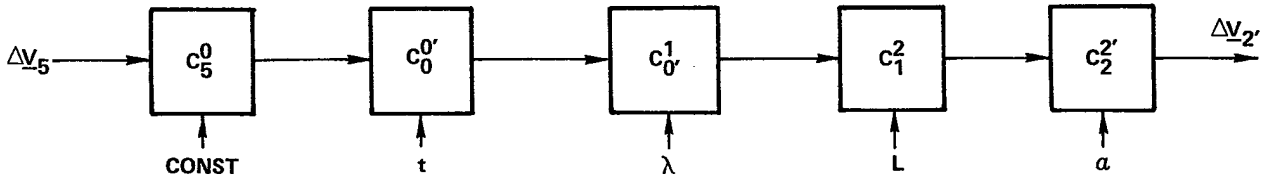


Figure 3-5

- STRAPDOWN INS

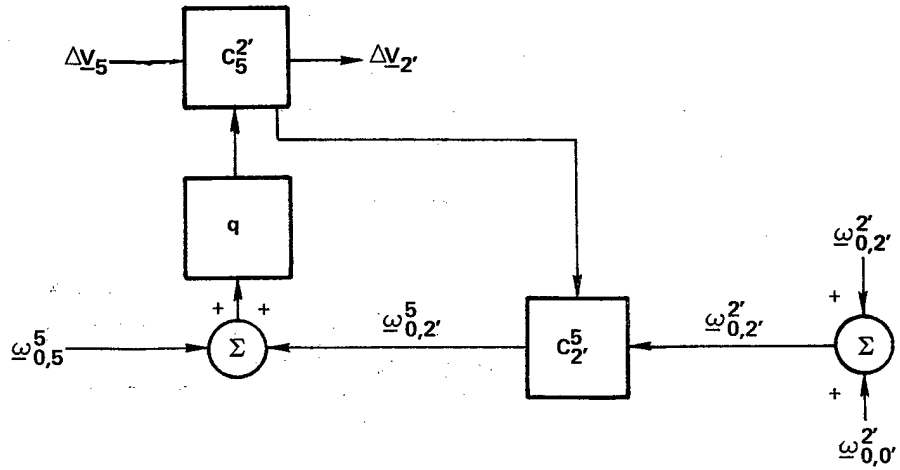


Figure 3-6

Figures 3-4, 3-5, 3-6. Additional Computations in the Local Geodetic Frame Required for Various INS Configurations - General Block Diagrams

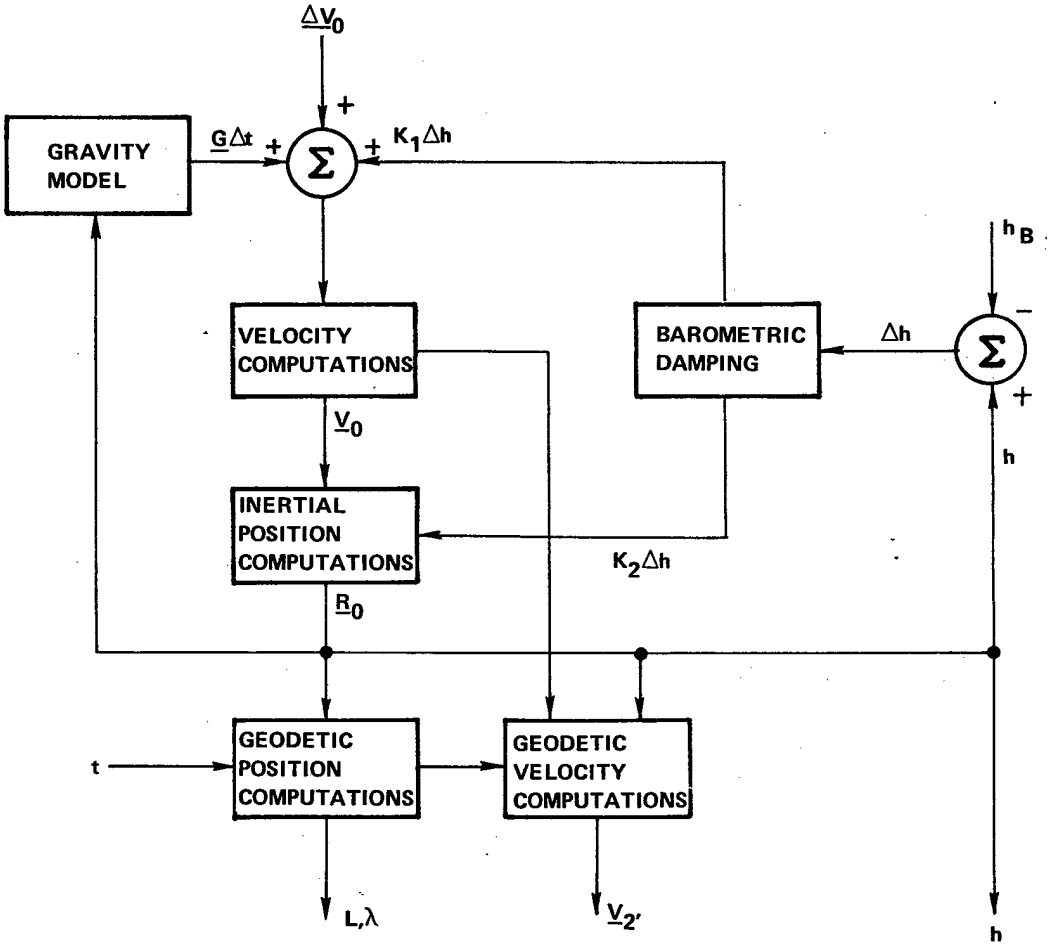


Figure 3-7. Commonality of Navigation Computations in a Space Stabilized Frame - General Block Diagram

to compute the local geodetic velocity components V_{x_2}, V_{y_2}, V_{z_2}. The $\underline{V}^0$ is also used with the barometric damping term, $K_2\Delta h$, in the computations of the local geodetic positon, components L, λ, h. The $\underline{R}^0$ is also used in the gravity computations and in the geodetic velocity computations.

Figure 3-8 presents additional computations in the space-stabilized frame required for an INS with a local level stabilized platform. The velocity increments, $\underline{\Delta V}^{10}$, sensed by the accelerometers mounted on the platform are transformed from the platform frame, 10, to the space-stabilized computational frame, 0, using the updated directional cosines matrix, $C_{10}^0 = C_{0'}^0 C_1^{0'} C_2^1 C_{10}^2$. In addition, the required gyro torquing signals are computed.

Figure 3-9 presents additional computations in the space-stabilized frame required for an INS with a space stabilized INS. The velocity increments, $\underline{\Delta V}^{10}$, sensed by the accelerometers mounted on the space stabilized platform, are transformed to the space stabilized axes frame, 0, using a constant transformation matrix C_{10}^0.

Figure 3-10 presents additional computations in space-stabilized frame required for a strapdown INS. The velocity increments $\underline{\Delta V}^5$ sensed by the accelerometers mounted on the aircraft body are transformed from the body axes frame to the computational frame using the direction cosine transformation matrix C_5^0. The matrix C_5^0 is computed from the quartenion, q. The quartneion is computed and updated using the angular velocity $\underline{\omega}_{0,5}^5$ sensed by the gyros.

3.6 Attitude Computations

Figures 3-11, 3-12, 3-13 present block diagrams of the aircraft attitude computations used either with the local-level computational frame or with the space-stabilized computational frame. The attitude angles are computed by solving the following matrix equation:

Attitude angles matrix = Transformation matrix from the aircraft body axes to the computation frame $C_5^{2'}$.

- INS WITH A LOCAL LEVEL STABILIZED PLATFORM.

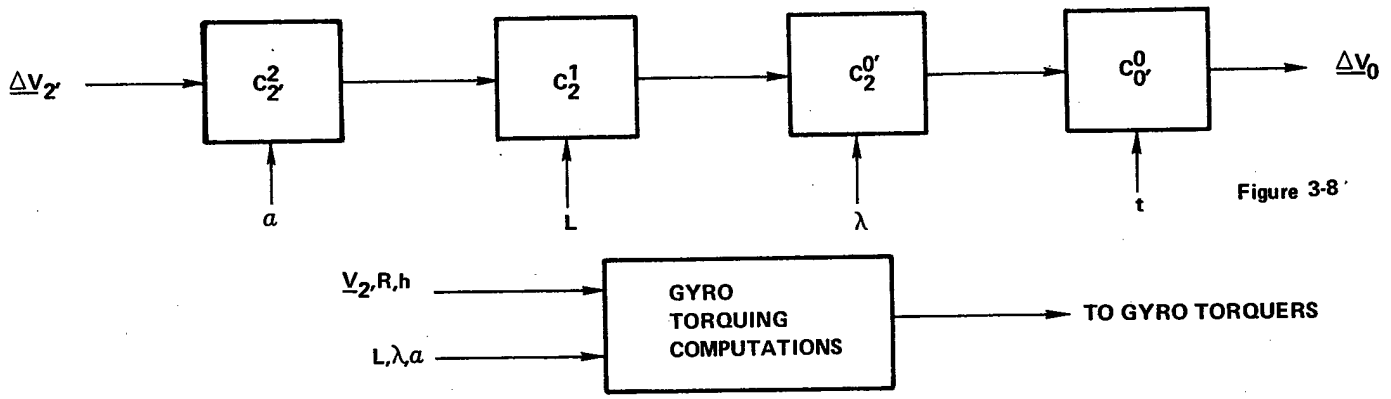


Figure 3-8

- INS WITH A SPACE STABILIZED PLATFORM.

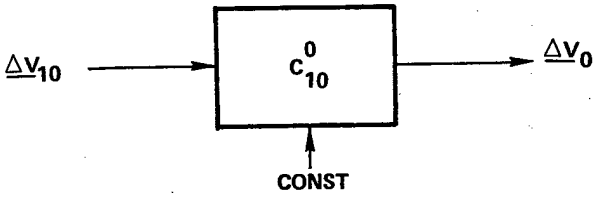


Figure 3-9

- STRAPDOWN INS.

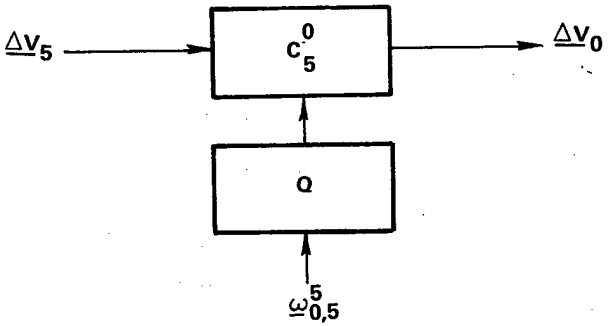


Figure 3-10

Figures 3-8, 3-9, 3-10. Additional Computations in the Space-Stabilized Frame Required for Various INS Configurations - General Block Diagram

- LOCAL LEVEL, GEODETIC, WANDER AZIMUTH INS

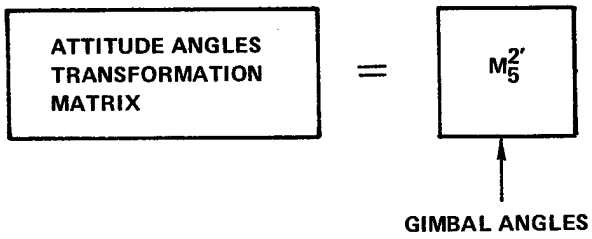


Figure 3-11.

- SPACE STABILIZED INS

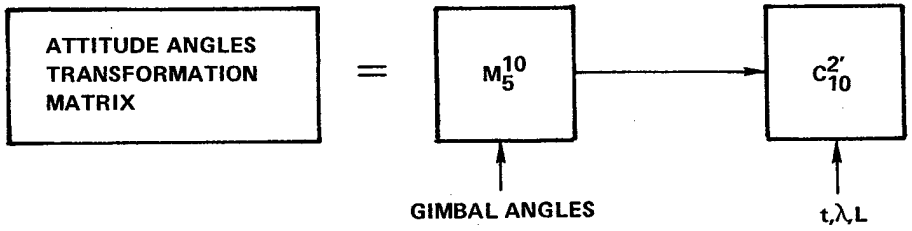


Figure 3-12.

- STRAPDOWN INS

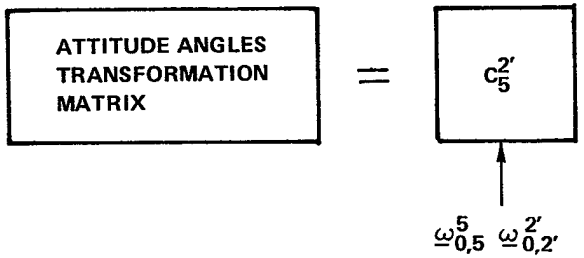


Figure 3-13.

Figures 3-11, 3-12 and 3-13. Attitude Computations - General Block Diagram

The attitude angles matrix is a function of the aircraft roll, pitch and a sum of yaw and wander angles. The transformation matrix, $C_5^{2'}$, for the gimballed type of the INS, with either the local-level (Figure 3-11) or the space-stabilized platform (Figure 3-12) consists of the transformation from the aircraft body axes frame to the platform axes frame followed by the transformation from the platform frame to the computational frame. The first matrix is computed from the gimbal angles and the second matrix is obtained from the navigation computations shown in Figures 3-3, 3-5, 3-7, 3-8.

For the strapdown INS (Figure 3-13) the transformation from the body axes frame, 5, to the local, wander-azimuth computational frame, 2', is obtained using the angular body rates with respect to the inertial frame, sensed by the strapdown gyros, $\underline{\omega}_{0,5}^{5}$, and the angular rate of the computational frame with respect to the inertial frame, $\underline{\omega}_{0,2'}^{2'}$, obtained from the navigation computations.

4.0 SURVEY OF SOFTWARE INTERFACES

4.1 Introduction

A survey of software interfaces between the INS navigation computations and other avionic system computations was performed to obtain technical data required for design of the computer errors simulation program, described in Volume III of this report. The survey results were used to obtain the following data.

a. The accuracy, the resolution and the data rate of the inertially sensed data required for the avionic computations.

b. The characteristics of the navigation aids measurements used in the INS navigation computations updating.

The following avionic computations are included in this survey:

a. A-7D INS Weapon Delivery and Flight Control Computations

b. A-7D INS Navigation Displays

c. A-7D INS Air Data Computations

d. C-5 INS Navigation Computations Updating

e. Electronically Agile Radar (EAR) Radar Antenna Motion Compensation Computations

The surveyed documentation was limited to the reports on file at CSDL at this time. The results of the survey are presented in Tables 4-1 to 4-7.

4.2 Summary and Conclusions

A survey of the software interfaces between the INS navigation computations and avionic systems computations summarized in Table 4-1, revealed typical accuracy. resolution and data rate of the inertially sensed data required for the avionic systems computations.

Table 4-1 Survey Summary of INS Software Interfaces With Avionic Computations

INS DESIGNATION	AVIONIC COMPUTATION	INERTIALLY DERIVED DATA		REQUIRED		
				RANGE	RESOLUTION (OR 1σ ACCURACY)	RATE
A-7D	WEAPON DELIVERY	INTEGRATED* SPECIFIC FORCE		10^{-2}g TO 10g	0.032 FT/SEC	25 HZ
		ATTITUDE		30 DEG/SEC IN PITCH AND YAW 200 DEG/SEC IN ROLL	2 ARC MIN	25 HZ
	AIRCRAFT FLIGHT CONTROL	INTEGRATED SPECIFIC FORCE		10^{-2}g TO 10g	0.032 FT/SEC	50 HZ TO 100 HZ
		ATTITUDE RATE	BODY PITCH RATE AND YAW RATE	30 DEG/SEC	0.01 DEG/SEC	ANALOG DATA WITH 15-30 HZ BANDWIDTH
			BODY ROLL RATE	200 DEG/SEC	0.05 DEG/SEC	
AN IMU MOUNTED ON EAR** ANTENNA	ANTENNA MOTION COMPENSATION (SAR***MAPPING)	INTEGRATED SPECIFIC FORCE		±2g	0.00125 FT/SEC	256 HZ
		ATTITUDE	LEVEL	20 DEG/SEC	0.6 ARC MIN	256 HZ
			AZIMUTH		2 ARC MIN	

* 10^{-4}g LOWER RANGE AND 0.00032 FT/SEC RESOLUTION USED DURING PRE-FLIGHT ALIGNMENT.
** ELECTRONICALLY AGILE RADAR
*** SYNETHETIC APERTURE RADAR

The inertially sensed data consists of linear acceleration, attitude and possible attitude rate. The velocity and position are derived from the inertially sensed data. The linear acceleration, data sensed by the accelerometers is in the form of pulses representing velocity increments. The attitude data is obtained from the gimbal angle readouts. The attitude rate data is obtained either by differentiation of the attitude data or it is sensed directly by the body mounted (strapdown) rate gyros.

The survey results indicate that severe resolution and rate requirements for the linear acceleration and the attitude data are imposed by the antenna motion compensation computations employed in the synthetic aperture radar operation (SAR), as shown in Table 4-1. The required velocity increments resolution is 0.00125 ft/sec with a data rate of 256 Hz for the acceleration range of $\pm$2g. The required level indication resolution is 40 arc-sec (1σ); the required azimuth resolution is 2 arc-min (1σ) and the required rate of the attitude data is 256 Hz. The data rate requirements are met by interpolating the inertial sensor data between 8 Hz navigation computation updates.

The A-7D navigation computations require a 0.032 ft/sec resolution of the velocity increments with a data rate of 5 Hz for an acceleration range of 10^{-2}g to 10g.

The A-7D weapon delivery computations require the velocity increment data in the navigation frame with a data rate of 25 Hz. The required attitude resolution is 2 arc-min (1σ) with a data rate of 25 Hz.

The A-7D flight control computations require velocity increment data in the aircraft body frame with a data rate of 50 Hz to 100 Hz for an acceleration range 10^{-2}g to 10g.

The A-7D navigation computations data requirements are sufficient for the display computations.

The A-7D inertially sensed data requirements are shown in Table 4-2.

The A-7D INS navigation computations use barometric altitude data to stabilize the vertical channel computations. The air data computer (ADC) is used to generate the barometric altitude data, the true and the indicated air speed data and the Mach number data. The true air speed data is used in computation of the wind velocity. The indicated air speed data is used for the head up display (HUD). The Mach number is used in the weapon delivery and in the automatic flight control computations. The air data range, accuracy and resolution are shown in Table 4-5.

A survey of the navigation computations updating by the navigation aids employed in C-5 INS indicates that the Kalman filter computations require 2200 words of 28 bits. The computations employ floating-point arithmetic with 28 bit words shared equally by mantissa and exponent. The sampling rate of the measurements is of the order of one sample per second and the Kalman filter updating cycle is of the order of minutes. The measurements samples are averaged over the Kalman computational cycle. The navigation aids measurements provide an update of the computed position, velocity and attitude. The measurements are also used to update the three gyro biases, the vertical accelerometer bias, the doppler radar measured drift angle, the computed ground speed scale factor and the barometric altimeter bias.

The range and accuracy of the typical navigation aids measurements employed in the aircraft navigation computations (not necessary in the C-5 INS) are shown in Table 4-6.

4.3 A-7D INS Weapon Delivery and Flight Control Computations

The weapon delivery and aircraft flight control computations require the following inertially derived data:

a. Linear Acceleration

b. Velocity

c. Position

d. Attitude

e. Attitude Rate

The inertial instruments sense directly the linear acceleration, the attitude and the attitude rate. The inertially sensed data is used to compute the velocity and the position. In some applications the attitude rate may be computed from the sensed attitude. In A-7D INS the attitude rate (the aircraft body rate) is obtained in analog form from the strapdown (body mounted) rate gyros.

Table 4-2 presents the range, the resolution and the rate of the inertially sensed data required for the A-7D weapon delivery and for the flight control computations.

4.3.1 Attitude Data

In A-7D INS the vehicle attitude is obtained from the gimbal angles measurements, providing the aircraft pitch, roll and yaw in relation to the geodetic local-level, north-pointing reference frame. The angle data is read out by synchros with 2 arc-min (rms) accuracy and it is used either directly or after A/D conversion in the avionic system computations. The attitude data is measured for the aircraft attitude changes up to 30 deg/sec in pitch and yaw and up to 200 deg/sec in roll. To prevent an excessive measurement error, the attitude data sampling rate 25 samples/sec is used. A summary of the typical attitude data utilization in A-7D avionic systems is shown in Table 4-3[17].

Table 4-2 Avionic Systems Use of Inertially Derived Data

INS TYPE	INERTIALLY DERIVED DATA		AVIONIC SYSTEM APPLICATION	REQUIRED INERTIAL DATA CHARACTERISTICS			REFERENCE	COMMENTS
				RANGE	RESOLUTION	RATE		
A-7D	AIRCRAFT BODY FRAME	PITCH	LISTED IN TABLE 4-3		0.03 DEG	25 HZ	17	ACCURACY AND RESOLUTION OF THE GIMBAL SYNCHRO READOUT IS 2 ARCMIN (1σ)
		ROLL			0.03 DEG			
		YAW			0.03 DEG			
		PITCH RATE	LISTED IN TABLE 4-4	30 DEG/SEC	0.01 DEG/SEC	15 HZ TO 30 HZ	17	
		ROLL RATE		200 DEG/SEC	0.05 DEG/SEC			
		YAW RATE		30 DEG/SEC	0.01 DEG/SEC			
	VELOCITY IN NAVIGATION FRAME		NAVIGATION	10^{-4}g TO 10g	0.032 FT/SEC PER PULSE	5 HZ	17	BELOW 10^{-2}g VELOCITY RESOLUTION IS 0.00032 FT/SEC
			WEAPON DELIVERY			25 HZ		
			DISPLAY					
	LINEAR ACCELERATION IN AIRCRAFT BODY FRAME		FCS PITCH CAS TURN COORD.	10^{-2}g TO 10g	0.032 FT/SEC PER PULSE	50 HZ TO 100 HZ	17	

Table 4-3 Typical Attitude Data Utilization in A-7D Avionic Computations[17]

MEASUREMENT OR COMPUTATION	APPLICATION	MEASUREMENTS REQUIRED	DATA RATE
NAV Frame to Body Frame Transformation	Target LOS Components transform ● All Bombing Modes Impact Point Components transformation ● Air-to-Gnd Gun/Rocket Mode ● Bomb Fall Line Gun Line Aim Angle Transformation ● Air-to-Air Gun/Rocket Mode FPM ● Velocity Components Transform Target El, Az Angle Transform ● ITV Bombing ● FLIR Bombing	Pitch, Roll, Yaw	25 25 25 25
Body Frame to NAV Frame Transformation	Target LOS Angle Transformation ● FLIR Bombing ● ITV Bombing Reticle El, Az Angle Transformation ● Visual Attack Bombing Mode ● Visual Attack Offset B M	Pitch, Roll, Yaw	25 *
Target LOS El Rate, Az Rate in Body Frame	Body Rate Compensation of FLIR and ITV	Pitch, Roll, Yaw	25
G Vector WRT Body Frame	Weapons delivery Computation	Pitch, Roll, Yaw	25
Sin & Cos of Roll	HUD display attitude positioning	Roll	
Pitch Angle	Pitch word to HUD	Pitch	25
Roll, Pitch Attitude	● Attitude Indication for VSD (ADI) ● Pitch and lateral steering commands from Flight Director Computer (FDC)	Roll, Pitch	*
Pitch, Roll Error Commands	FCS attitude hold error commands for autopilot	Pitch, Roll	*
Heading	Steering Commands for VSD, HSD, HUD	Pitch, Roll, Yaw	*
Attitude	FCS attitude compensation calculations	Pitch, Roll, Yaw	*

*ANALOG DATA

4.3.2 Attitude Rate Data

In A-7D the aircraft attitude rate data is obtained from the strapdown gyros that are a part of the Flight Control System (FCS). The data is in analog form with 15 to 30 Hz bandwidth. Resolution of the pitch and yaw rate gyros data is 0.01 deg/sec and resolution of the roll gyro data is 0.05 deg/sec. The FCS computations including the computations for the pitch and roll axes Control Augmentation System (CAS) and the yaw axis Stability Augmentation System (SAS) are the principal users of the attitude rate data. Outside the FCS the attitude rate data is required for the Forward-Looking Radar (FLR) computations when operating in terrain-following or terrain-avoidance mode. The atittude rate data is used to obtain the turn rate information for offset of the azimuth pointing of the radar antenna during the turn to indicate to the pilot where he is going. The turn rate is also displayed on the Attitude Director Indicator (ADI). The aircraft pitch and yaw rates could be also used in computation of gun-line lead angle for air-to-air weapon delivery at fixed range using a servoed aiming vehicle which the pilot maintains on target.

A summary of the representative type of the avionic computations utilizing the attitude rate data is shown in Table 4-4[17].

4.3.3 Linear Acceleration Data

The linear acceleration (specific force) is sensed by the torque-restrained, pendulum type accelerometers, mounted on the IMU platform. The accelerometer output pulses represent velocity increments at 0.032 ft/sec/pulse (low-gain scaling, used during the flight). The overall specific forces range sensed by the accelerometers is from 10^{-4}g to 10g. The required data rates are 5 Hz for the navigation computations, 25 Hz for the weapon delivery computations and 50 Hz to 100 Hz for the FCS computations.

Table 4-4 Typical Attitude Rate Data Utilization in A-7D Avionic Computations[17]

Measurement or Computation	Application	Data Form Required	DATA RATE (Samples/sec)
Body Pitch Rate	FCS: Pitch axis CAS	Body pitch rate	*BW: 15–30 cps
Body Roll Rate	FCS: Roll axis CAS	Body roll rate	*BW: 15–30 cps
Body Yaw Rate	FCS: Yaw axis SAS	Body yaw rate	*BW: 15–30 cps
Turn Rate	FLR offset azimuth pointing during turn	Body yaw rate	* 25
Turn Rate	VSD indication	Body yaw rate	* 5
Aiming Lead Angle	Gun-Line Lead angle	Body pitch & yaw rate	25
Body attitude Rate Compensation	Rate-aided FLR attitude stabilization	Platform Pitch Gimbal Rate; Platform Roll Gimbal Rate or Body Pitch rate Body Yaw rate	
Body attitude Rate Compensation	Rate-aided boresight stabilization for FLIR and ITV	Body pitch rate Body yaw rate	

*ANALOG DATA

In the A-7D, acceleration data for navigation functions is accumulated and corrected for Coriolis, centripetal, and gravity accelerations to yield a high-accuracy velocity update five times a second. The weapon delivery calculations, however, require more frequent velocity information. To satisfy this latter requirement, the raw accelerometer data is sampled 25 times a second, it is corrected for gravity acceleration and it is added incrementally to the latest velocity update, thus providing five velocity interpolations between each navigation velocity update.

For the FCS computations the acceleration data is sampled at a 50 to 100 samples per second rate and it is tranformed from the platform frame to the aircraft body frame.

4.4 A-7D Navigation Displays Computations

The following major navigation displays are employed in A-7D:

1. Attitude Director Indicator (ADI)
2. Horizontal Situation Indicator (HSI)
3. Head-up Display (HUD)
4. Forward Looking Radar (FLR)
5. Projector Map Display System (PMDS)

A listing of parameters presented in each display is given below

ADI

Roll, Pitch Attitude
Magnetic Heading
Horizontal, Vertical Steering Pointers
Displacement Pointer
Slip Indication
Rate-of-turn Indicator

HSI

Range Display
Compass Card
Settable Heading Marker
Course Pointer and Settable Digital Course Readout
To - From Arrow
Course Deviation Bar

HUD

Flight Path Angle, Marker
Horizon Line
Angle-of-attack
Pull Up Command
Airspeed
Altitude/Vert Velocity
Radar Altitude Advisory Symbol
Magnetic Heading
Flight Director
Landing Director
Terrain Following Pitch Steering Commands
Drift Angle
Glide Slope Deviation
Localizer Deviation

FLR

5" CRT Sector Display
Adjustable Cursor for NAV Marks

Projected Map Display

5" Projected Slewable Map Sector Display
Present Position
Magnetic Track Angle
Bearing to Destination
Range to Destination
Landmark Update Capability

The accuracy, the resolution and the data rate of the inertially derived parameters required for the navigation and the weapon delivery computations are adequate for the navigation displays.

4.5 A-7D Air Data Computations

The A-7D air data computer (ADC), CP-953A/AJQ uses the sensed static pressure, the ram pressure and the free air temperature to compute the true air speed, the indicated air speed, the Mach number and the barometric attitude.

The true air speed is used with the inertially derived velocity, heading, attitude and with the angle of attack (AOA) from the AOA transducer in computation of the wind velocity. The computed wind velocity is stored for use in the weapon delivery computations, in the degraded navigation modes of operation and it is displayed for the pilot.

The indicated air speed is used for the pilot head-up display (HUD).

The Mach number is used in the weapon delivery computations and in the automatic flight control computations.

The barometric altitude is used primarily to stabilize the vertical channel inertial navigation computations. In addition, the barometric altitude is used in computations employed in FLR (in the sweep delay correction for the ground map use), in AFCS (in attitude hold computations), in IFF/SIF (in altitude reporting) and in AAU-19/A altimeter (in pilot readout).

Table 4-5 presents a list of the ADC derived data including the range, the accuracy, the resolution and the data format [18].

Table 4-5 Air Data Computer Measurements[18]

INS TYPE	AVIONIC SYSTEM TYPE	AVIONIC SYSTEM DERIVED DATA	PRIMARILY APPLICATION	AVIONIC DATA CHARACTERISTICS				REFERENCE
				RANGE	ACCURACY	RESOLUTION	SIGNAL TYPE & VOLTAGE LEVEL	
A-7D	AIR DATA COMPU-TER	TRUE AIRSPEED	COMPUTATION OF WIND VELOCITY	118.5 TO 600.2 KNOTS	4 KNOTS BELOW 202 KNOTS 4.74 KNOTS ABOVE 202 KNOTS	1 KNOT	DC ANALOG 0 TO 4V	18
		MACH NUMBER	WEAPON DELIVERY AND AUTOMATIC FLIGHT CONTROL	0.20 TO 96 MACH	0.008 MACH BELOW 30,000 FT 0.01 MACH ABOVE 30,000 FT	0.002 MACH	SYNCHRO 11.8 VAC	18
		BARO-METRIC ALTITUDE	STABILIZATION OF VERTICAL CHANNEL COMPUTATIONS	−1,000 TO 50,000 FEET	25 FT OR 0.25% OF ALTITUDE	<25 FT AT POTENTIO-METER OUTPUT	DC ANALOG 0 TO 4V	18
		INDICATED AIRSPEED	HUD (DISPLAY)	SAME AS TRUE AIRSPEED				

4.6 C-5 INS Navigation Computations Updating

The C-5 Kalman Navigation/Alignment Mechanization is shown in functional block diagram [19], Figure 4-1. The INS computations are updated by the navigation and measurements using Kalman filter. The updating provides corrections to the following data:

Position (2)
Velocity (2)
Altitude Rate (1)
Altitude (1)
Computed Heading (1)
Platform Leveling Rates (2)
Gyro Biases (3)
Vertical Accelerometer (1)
Doppler Drift Angle (1)
Ground Speed Scale Factor (1)
Barometric Altitude (1)

The numbers in brackets shown above indicate the number of components updated by the Kalman filter computations.

The total number of updated variables (state variables) is sixteen.

The following navigation aids are employed in the updatings [19].

1. Doppler Radar
2. Loran Receiver
3. Tacan Receiver
4. Multimode Radar
5. Position Fix
6. Barometric Altimeter

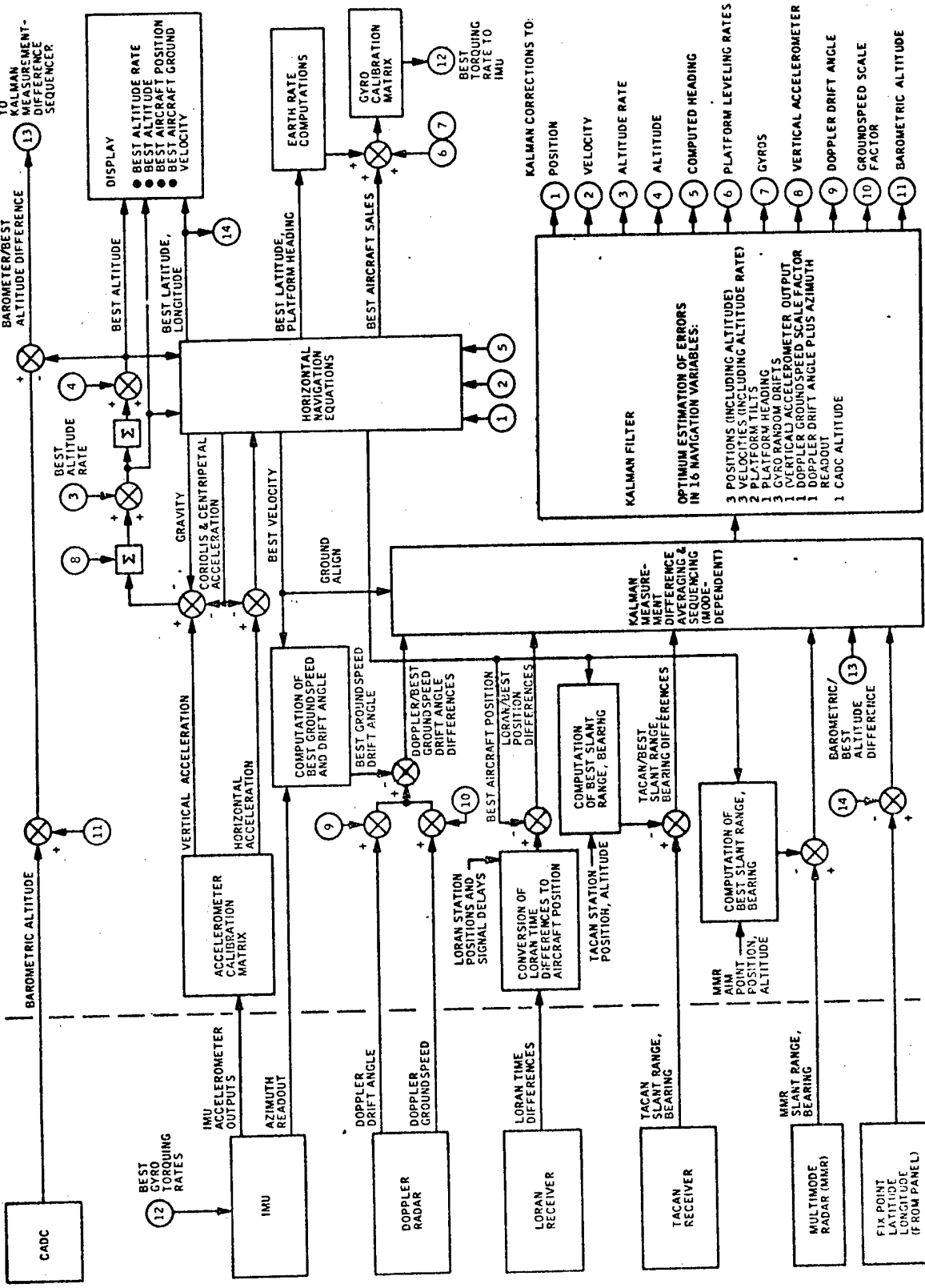


Figure 4-1. C-5 Kalman Navigation/Alignment Mechanization, Functional Block Diagram(5)

- The Doppler radar, manufactured by the GPL Division of General Precision Systems, Inc., supplies the computer with measurements of ground speed and drift angle. By utilizing the information from the ball readout, it is possible to obtain the ground speed relative to platform axes.

- The Loran receiver is manufactured by Collins Radio Company. When available, Loran data is converted in digital computer to provide position information displayed to the navigator. Subject to the navigator's judgment and control, the Loran data can also be used to update the best estimate of the state vector which includes position, velocity, and other quantities.

- The Tacan receivers, manufactured by Hoffman Electronics, provide slant range and bearing information to ground stations at known locations. The navigator can use this data, when available, in a manner similar to that described for the Loran.

- The multimode radar (MMR), manufactured by Norden Division of United Aircraft, provides many modes of operation for the C-5. Two modes are used to obtain navigation information. In the terminal phase of flight, the navigator can, through a cathode ray tube display, sight on landmark patterns whose position relative to the desired supply and/or personnel drop location is known. Slant range and bearing data is provided by the MMR during the terminal phase. The navigator can also use the MMR for sighting on landmarks whose absolute position (latitude and longitude) are known. In this mode, measurements of range, bearing and landmark location go directly to the Kalman filter.

- Position fix data are available from Loran, Tacan, multimode radar, visual updates and celestial fixes. When the operator requests position fix data from Tacan, for exaple, the Tacan routine processes the receiver's slant-range and bearing information into craft position. Differences between the system's best estimate of craft position and the Tacan determined craft position are computed and displayed to the operator along the Tacan determined position and a suitable radial position difference. In addition, a statistical reasonableness test is performed on each of the Tacan mea-

surements, and the operator is informed of the results. If the processor finds the fix to be reasonable, the operator may accept it (FIX ENABLE) or reject it (FIX REJECT). If the processor finds the fix to be unreasonable, the operator may reject it or override it (FIX OVERRIDE). This same procedure is followed when the operator requests data from the Loran, multimode radar, visual fixes or celestial fixes.

- The barometric altimeter is a part of the two air data computer (CADC) manufactured by Elliot Brothers. The barometric pressure altitude data is used in the navigational computations.

- Kalman filter computations are mechanized on NDC-1051A digital computer, manufactured by Northrop Electronics Division. The sampling rate of the navigation aids measurements is of the order of one sample per second and a typical Kalman computation updating cycle is of the order of a few minutes. The measurements samples are averaged over the Kalman computation cycle. The computations employ floating-point arithmetic utilizing 28-bit word shared equally by mantissa and exponent with 2200-words allocation.

The range and accuracy of typical navigation aids measurements employed in the aircraft navigation (not necessarily in C-5) are shown in Table 4-6[20] [21].

Table 4-6 Typical Characteristics of Navigation Aids Measurements

INS TYPE	AVIONIC SYSTEM TYPE	AVIONIC SYSTEM DERIVED DATA	INERTIAL SYSTEM NAVIGATION APPLICATION	TYPICAL NAV-AIDS MEASUREMENTS CHARACTERISTICS		REFERENCE
				RANGE	ACCURACY (1σ)	
AN AIDED INS (C-5)	DOPPLER RADAR	DRIFT ANGLE	KALMAN FILTER COMPUTATION OF CORRECTIONS LISTED IN SECTION 4.6	30 DEG	0.1 DEG	20
		GROUND SPEED		100 TO 1000 KNOTS	0.1% + 0.1 KNOT	
	LORAN C RECEIVER LORAN D RECEIVER	AIRCRAFT POSITION		MAJOR SHIPPING LANES	350 FT	21
	TACAN RECEIVER	SLANT RANGE TO TACAN STATION		300n MILES	600 FT	
		BEARING TO TACAN STATION			1 DEG	
	MULTI MODE RADAR	SLANT RANGE TO KNOWN POINT				
		BEARING TO KNOWN POINT				
	POSITION FIX	LATITUDE LONGITUDE				
	BARO-METRIC ALTIMETER	BAROMETRIC ALTITUDE		−1000 TO 50,000 FT	5 FT OR 0.05% OF ALTITUDE	20

4.7 SAR Antenna Motion Compensation

The Synthetic Aperture Radar (SAR), used for mapping to the side of the aircraft, is a part of the Electronically Agile Radar (EAR). The required mapping azimuth resolution is obtained in SAR mapping by generation of an effective long antenna derived from signal processing means rather than from the actual use of a long physical antenna. In SAR a radiating element, translated with the aircraft, assumes sequential positions along a line. At each of these positions a signal is transmitted and the amplitude and phase of the radar signals reflected from the terrain are stored. After the radiating element transversed a predetermined distance, equal to the effective length of the antenna, the stored signals resemble strongly the signals that would have been received by the elements of an actual linear array with the same effective antenna length. Consequently, the effect of a long physical linear antenna array is achieved by a single moving radiating element.

An uncompensated motion of the SAR antenna during the mapping time interval is introducing phase errors in the stored radar signals. The resultant range errors are directly proportional to the phase errors.

The SAR antenna motion may be sensed by an IMU, hard-mounted on the antenna. Two possible IMU configurations employed for the antenna motion compensation are as follows:

1. A high quality master IMU mounted directly on the antenna.

2. A less accurate, slave IMU mounted on the antenna and a high quality master IMU mounted approximately 1 meter away.

The antenna-mounted IMU performance required for the SAR antenna motion compensation is listed in Table 4-7.

Table 4-7 IMU Data Requirements for EAR Antenna Motion Compensation

INS TYPE	INERTIALLY DERIVED DATA	AVIONIC SYSTEM TYPE	AVIONIC SYSTEM APPLICATION	REQUIRED INERTIAL DATA CHARACTERISTICS				REFERENCE
				RANGE	ACCURACY (1σ)	RESOLUTION	RATE	
IMU MOUNTED ON THE RADAR	LEVEL INDICATION (ROLL, PITCH)	EAR	SAR MAPPING TO THE SIDE OR THE AIR-CRAFT	20 DEG/SEC	0.01 DEG		256 HZ	22
ANTENNA	AZIMUTH			20 DEG/SEC	0.03 DEG		256 HZ	22
	ACCELERO-METER OUTPUT (VELOCITY INCREMENTS)			$\pm$2g ACCELER-ATION		0.00125 FT/SEC	256 HZ	22

The listed accuracy, resolution and rate of the inertial data is required to meet the present design criteria for the SAR [22] the assumptions used in developing this specification are:

1. The IMU is hard mounted to the antenna and the antenna is hard mounted to the aircraft.

2. B-1 aircraft environment with a 2 g maneuver and angular rates of 20 deg/sec.

3. Mapping range 50 n miles, the antenna squint angle 20 deg, the aircraft velocity 600 ft/sec.

4. Hamming weighting the Peak Side Lobe Ratio (PSLR) is below -30dB, the Strehl intensity is larger than 0.8 and the flare ratio (used in specifcation of the average clutter) is 0.1.

SECTION 5

SELECTION OF COMPUTATIONAL FRAME FOR THE COMPUTER ERRORS SIMULATION PROGRAM

The INS navigation computations may be performed in either the local-level or the space-stabilized frame.

The criteria used in the selection of the computational frame for the computer errors simulation program are as follows:

1. World-wide navigation computations capability.
2. Maximum commonality in the navigation computations for the INS with a local-level stabilized platform, a space-stabilized platform and a strapdown configuration.
3. Minimum of required computer complexity.
4. Simple interface with the avionic systems' computations including the computations used for the flight control, weapon delivery, and the radar antenna motion compensation. Simple interface with the navigation aid measurements data.

The local-level, geodetic wander-azimuth frame was selected as a computational frame for the INS computer errors simulation program. The frame selection was dictated primarily by the simpler computation format for the INS with a local-level stabilized platform and simpler software interfaces with most avionics computations and navigation-aid measurements. Computations in the space-stabilized frame are, however, somewhat simpler for the space-stabilized and strapdown INS.

Comparison of the local-level and the space-stabilized computational frames characteristics is summarized in Table 5-1.

Table 5-1. Comparison of Local-Level and Space-Stabilized Computational Frames

	COMPUTATIONAL FRAME CHARACTERISTICS	
SELECTION CRITERIA	LOCAL-LEVEL	SPACE-STABILIZED
World-Wide Navigation	Requires Wander-Azimuth, Free-Azimuth, Unipolar, or Inverse Coordinates	No Added Requirements
Commonality of Computations	Required Additional Transformations in Attitude Computations for Space-Stabilized and Strapdown INS	See Additional Transformations Required
Complexity of Computations	Simpler Computations for Local-Level INS	Simpler Computations for Space-Stabilized and Strapdown INS
Interface with Other Avionic Computations	Simpler Interface with Most Avionics and Navigation Aids Computations	Simpler Interface with Some (Star Tracker) Navigation Aids Computations

The remaining part of this section contains a discussion of the frame selection criteria listed above.

1. World-wide Navigation Capability. Near the earth's poles, latitude-longitude are poor navigation coordinates because of the convergence of meridians. This effect results in both mechanical and computational difficulties in the implementation of the local-level platform stabilization. For the INS with the north-pointing platform stabilization operating near the polar region, the required azimuth gyro torquing rate becomes excessive and the computation of certain latitude functions and longitude rate have singularities at the poles.

These difficulties could be removed by the use of modified local-level coordinates for the platform stabilization frame and for the navigation computational frame. Various modified local-level frames employ the wander-azimuth, the free-azimuth, the unipoloar, and the inverse coordinates.

The navigation computations in the wander-azimuth frame, selected for this simulation program, determine the INS position and velocity in the latitude-longitude coordinates. While the wander-azimuth frame computations eliminate the problems encountered in the computations utilizing the local-level north-pointing frame, the use of latitude-longitude coordinates for the polar region navigation is not the best choice.

The navigation computations in the space-stabilized frame determine the INS position and velocity in the space-stabilized coordinates. The computed position and velocity can be then resolved in any desired frame (for example along the track, cross track and altitude directions).

2. Maximum Commonality of the Navigation Computations. The commonality of computations for both computational frames starts from processing of the velocity increments sensed by the accelerometers and resolved in the computational frame axes and ends with the computed position and velocity resolved in the desired coordinates, as shown in Figures 3-3 and 3-7.

The aircraft attitude computations have the same format for either of the computational frames, as shown in Figures 3-11, 3-12, and 3-13. However, for the computations in the local-level wander-azimuth computational frame, additional transformations are required for either the space stable or the strapdown INS due to the absence of the inertial frame and the inclusion of the wander angle.

3. Minimum of Required Computer Complexity. For the INS with a local-level stabilized platform the navigation computations in the local-level frames are somewhat simpler than the navigation computations in the space-stabilized frame.

For the INS with a space-stabilized platform and for the strapdown INS, the navigation computations in the space-stabilized frame are considerably simpler than the navigation computations in the local-level frame.

4. Simpler Interface with the Avionic System Computations and with the Navigation Aids Measurements. The navigation computations in the local-level wander-azimuth frame have somewhat simpler interface with some avionic system computations and with the navigation aids measurements than the navigation computations in the space-stabilized frame.

6.0 SELECTION OF STANDARDIZED FRAMES SYMBOLOGY.

6.1 INTRODUCTION

This section presents a proposed standardized symbology of the coordinate frames and the frames transformations employed during the INS navigation computations. It is recognized that the proposed standardization is somewhat arbitrary and incomplete, but it represents a reasonable simple and consistent sets of symbols and definitions of frame axes orientations. The proposed coordinate frames symbology is based on terminology employed by Honeywell in GEANS studies[2] and the proposed transformation frames symbology is based on Britting notation[15]. A tutorial discussion of transformation of coordinates, described in Appendix A, is used as a background for the material presented in this section.

6.2 PROPOSED SYMBOLOGY STANDARDIZATION

1. A right-handed, orthogonal, cartesian coordinate frame, i, is defined by specifying three orthogonal unit vectors $\hat{x}_i$, $\hat{y}_i$, $\hat{z}_i$ along the frame axes.
2. A vector $\underline{A}$ expressed in the frame i, is designated by $\underline{A}^i$. The components of the $\underline{A}$ vector in the i frame are designated by A_{x_i}, A_{y_i}, A_{z_i}.
3. A vector $\underline{B}$ representing a motion of the frame i with respect to the frame j and expressed in the frame k is designated by $\underline{B}^k_{j,i}$.
4. A coordinate transformation from the frame i to the frame j, having a common origin, is represented by angles of rotation about at most the three i frame exes required to bring the frame i axes to coincide with the corresponding frame j axes.

For example the following rotations may be required to bring the frame i axes into coincidence with the frame j axes:

a. A rotation about the x_i axis through an angle α.

b. A rotation about the y_i axis (displaced by the previous rotation) through an angle β.

c. A rotation about the z_i axis (displaced by the previous two rotations) through an angle γ.

The angles α, β, γ are called the Euler angles. The positive sign of the rotation angles is defined by a right-hand rule. An example of a positive rotation by an angle α about the x_i axis is shown in Figures 6-1 and 6-2.

The rotation angles are usually expressed by a direction cosine matrix (9 terms) or a quaternion (4 terms). The coordinate transformation from the frame i to the frame j expressed by a direction cosine matrix is designated by C_i^j and expressed by a quaternion is designated by q_i^j.

5. The major coordinate frames used in the INS navigation computations are shown in Figures 6-3, 6-4 and they are listed in Table 6-1.

6. A general circle diagram representing the coordinate frames transformations used in the navigation computations of a local-level, a space-stabilized and a strapdown INS is shown in Figure 6-5. The transformations are also listed in Table 6-2.

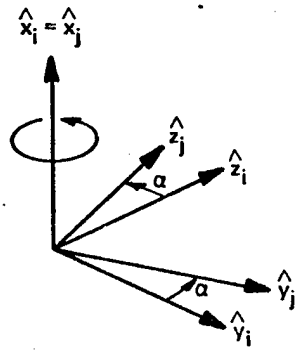


Figure 6-1. Rotation About X-axis, Vector Representation

"from" frame | axis of rotation | "to" frame

Figure 6-2. Rotation About X-axis, Circle Diagram Representation

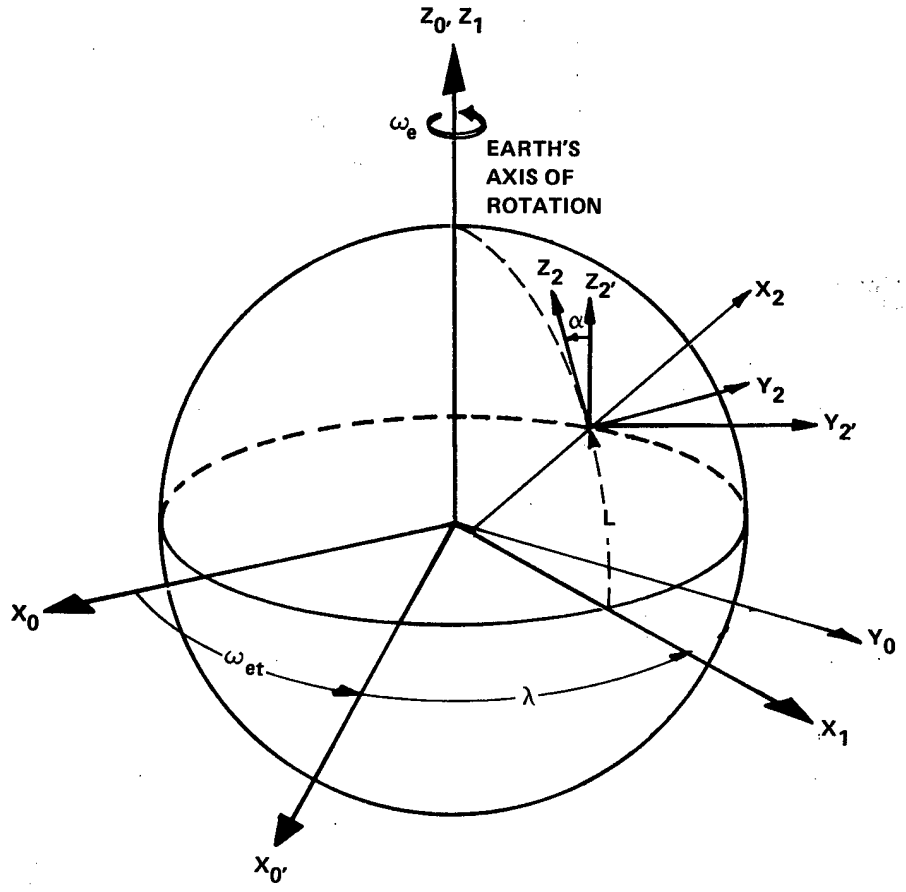


Figure 6-3. Frames Used in SPN/GEANS Computations

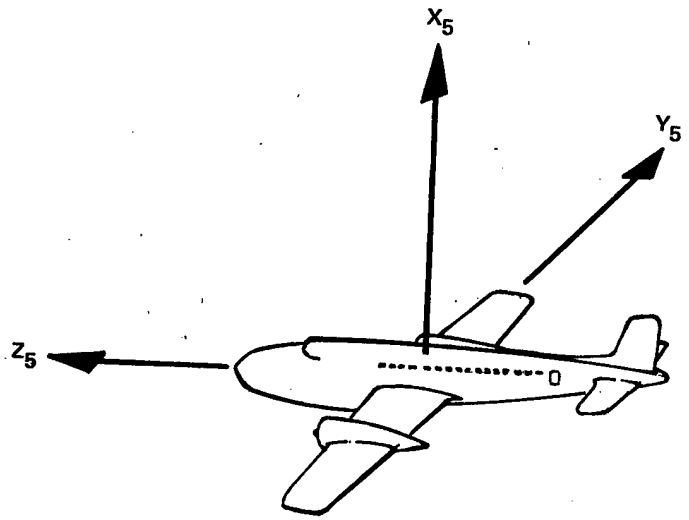


Figure 6-4. Vehicle Frame Axes Used in SPN/GEANS Computations

Table 6-1. Coordinate Frames of Interest for SS, SD, and LVWA Systems with SS or LVWA Navigation

0–frame	THE INERTIAL REFERENCE FRAME defined at moment of entry into navigation i.e. at t = 0.
0′–frame	THE EARTH FIXED FRAME coincides with 0–frame at t = 0.
2–frame	THE LOCAL VERTICAL NORTH FRAME is the frame of the position and velocity outputs.
2′–frame	THE LOCAL VERTICAL WANDER AZIMUTH FRAME coincides with stable element of LVWA platform and generally with 2–frame at t = 0.
5–frame	THE BODY OR STRAPDOWN PLATFORM FRAME defined by lateral and longitudinal axes of the vehicle.
10–frame	THE SPACE STABLE PLATFORM FRAME coincides with stable element of SS platform (any orientation).

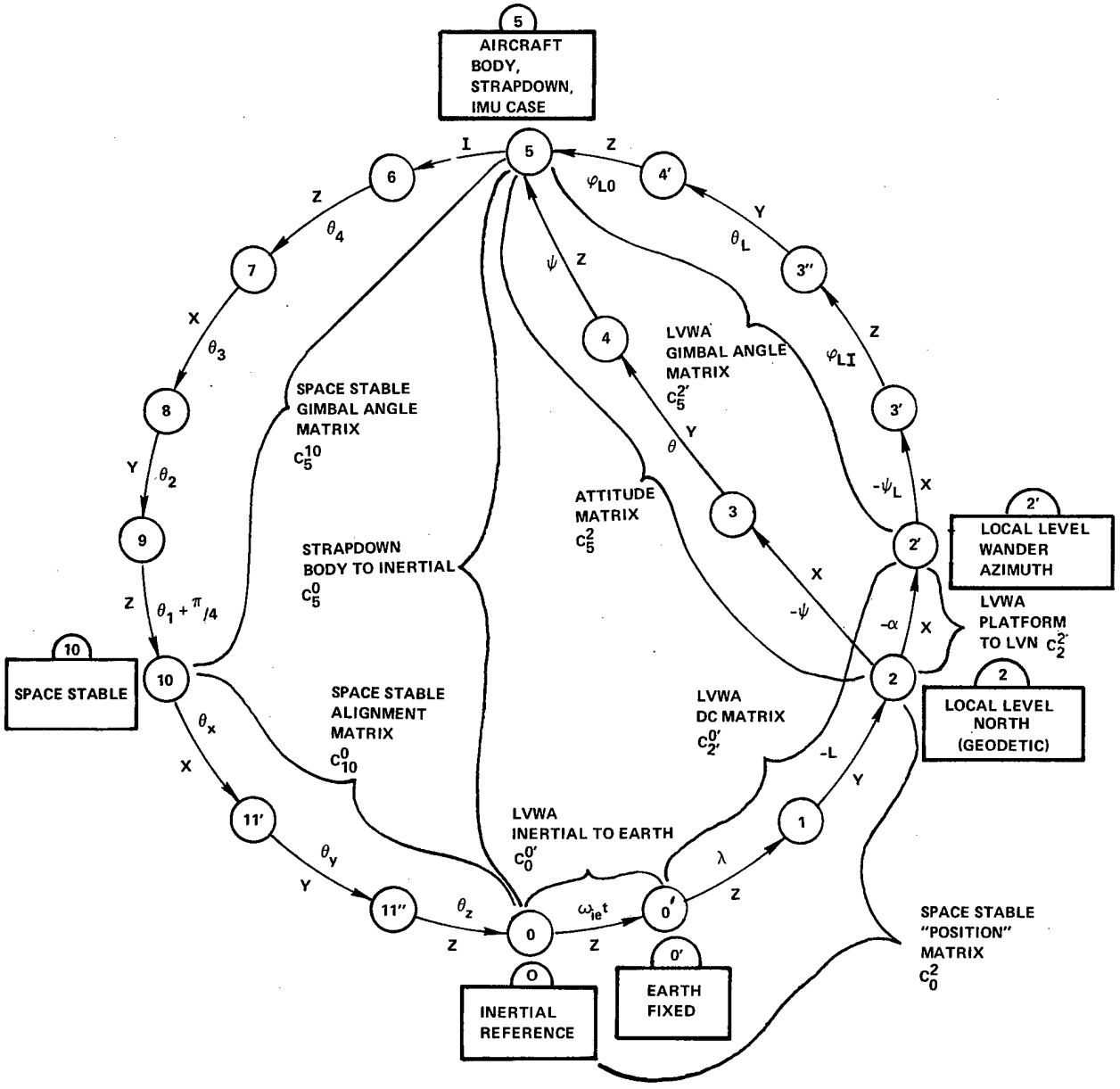


Figure 6-5. General Circle Diagram for SS, SD, LVWA INS Computations

TABLE 6-2. COORDINATE FRAMES APPEARING IN GENERAL CIRCLE DIAGRAM

FRAME DESIGNATION		FRAME AXES ORIENTATION			COORDINATES TRANSFORMATION			COMMENTS	TRANSFORMATION MATRIX NAME
NAME	CODE NUMBER	X	Y	Z	ROTATION AXIS	ROTATION ANGLE SYMBOL	ROTATION ANGLE NAME		
INERTIAL REFERENCE (EARTH CENTERED)	0	IN EQUATORIAL PLANE THROUGH INS MERIDAN AT t=0	90° EAST OF X - AXIS IN THE EQUATORIAL FRAME	EARTH'S AXIS OF ROTATION (UP AT THE NORTH POLE)					
EARTH FIXED	0'	SAME AS ABOVE FOR t=0	SAME AS ABOVE FOR t=0	SAME AS ABOVE	Z	$\omega_{\lambda_{ie}t}$	EARTH RATE TIMES TIME	THE NAVIGATION COMPUTATION STARTS AT t=0	INERTIAL TO EARTH FIXED MATRIX
	1	SAME AS ABOVE FOR λ=0	SAME AS ABOVE FOR λ=0	SAME AS ABOVE	Z	λ	CHANGE IN LONGITUDE	λ IS POSITIVE EAST	LVWA DCM
LOCAL VERTICAL, NORTH POINTING	2	UP	EAST	NORTH	Y	$-L$	GEODETIC LATITUDE	L IS POSITIVE NORTH	
LOCAL VERTICAL WANDER AZIMUTH	2'	UP	EAST FOR α = 0	NORTH FOR α = 0	X	$-\alpha$	WANDER ANGLE	α IS POSITIVE FOR NORTH TO EAST ROTATION ABOUT UP AXIS	
INNER ROLL GIMBAL	3'	UP	EAST FOR α = 0, ψ_L = 0	NORTH FOR α = 0, ψ_L = 0	X	$-\psi_L$	YAW	SIGN OF ROTATION SAME AS FOR α	GIMBAL ANGLE MATRIX FOR LVWA PLATFORM
PITCH GIMBAL	3''	UP FOR ϕ_{LI} = 0	EAST FOR α = 0, ψ_L = 0, ϕ_{LI} = 0	SAME AS ABOVE	Z	ϕ_{LI}	INNER ROLL	NOMINALLY = 0	
					Y	θ_L	PITCH	POSITIVE FOR NOSE UP	
OUTER ROLL GIMBAL	4'	UP FOR ϕ_{LI} = 0, θ_L = 0	SAME AS ABOVE	FUSELAGE FORWARD	Z	ϕ_{LO}	OUTER ROLL	POSITIVE FOR RIGHT WING DOWN	
AIRCRAFT BODY OR PLATFORM CASE	5	UP (THROUGH THE CANOPY)	RIGHT WING	FUSELAGE FORWARD				IDENTITY FOR THIS STUDY	SS GIMBAL ANGLE MATRIX
SS IMU CASE	6	SAME AS ABOVE			Z	θ_4			
SS GIMBAL #4	7				X	θ_3			
SS GIMBAL #3	8				Y	θ_2			
SS GIMBAL #2	9				Z	$\theta_1 + \pi/4$			
SS PLATFORM	10				X	θ_X			SS ALIGNMENT MATRIX
	11'				Y	θ_Y			
	11''				Z	θ_Z			
INERTIAL REFERENCE (EARTH CENTERED)	0	D E F I N E D A B O V E							SS POSITION MATRIX
	1								
LOCAL VERTICAL NORTH POINTING	2								
PITCH GIMBAL FOR NORTH POINTING, LV, 3 GIMBAL IMU	3	UP	EAST FOR ψ = 0	NORTH FOR ψ = 0	X	$-\psi$	HEADING	ψ IS POSITIVE EAST	ATTITUDE MATRIX
ROLL GIMBAL FOR NORTH POINTING, LV, 3 GIMBAL IMU	4	UP FOR θ = 0	EAST FOR ψ = 0	NORTH FOR ψ = 0, ϑ = 0	Y	θ	PITCH	θ IS POSITIVE NOSE UP	
AIRCRAFT BODY	5	UP (THROUGH THE CANOPY)	RIGHT WING	FUSELAGE FORWARD	Z	ϕ	ROLL	ϕ IS POSITIVE FOR RIGHT WING UP	SD BODY TO INERTIAL MATRIX

REFERENCES

VOLUME II

1. Seppelin, T. O., The Department of Defense World Geodetic System, 1972, Defense Mapping Agency, May 1974.

2. Computer Program Development Specification for the Standard Precision Navigator/GEANS Inertial Navigation System (Configuration Item Number DQG8156A1) 4th Preliminary Issue, (no author named), Honeywell, Inc., St. Peterburg, Fla., 1 September 1974.

3. Crews, L. L. and Hall, C.W., A-7D/E Aircraft Navigation Equations, Technical Note 404-176, Naval Weapons Center, China Lake, Cal., March 1975.

4. Widnall, W.S. and Grundy, P.A., Inertial Navigation System Error Models, TR-03-73, Intermetrics, Inc., Cambridge, Ma., 11 May 1973.

5. B-1 Offesnsive Flight Software -- Computer Program Implementation Concept, Code Identification Number 81205, (no author named), Boeing Co., Seattle, Wash., 1973.

6. H-386 System Equations and Mechanization, Revision B, (no author named), Honeywell, Inc., St. Petersburg, Fla., 1 August 1966.

7. Davenport, W., FAWN Free Azimuth World-Wide Navigation, Honeywell, inc., St. Petersburg, Fla., (Internal memorandum), 13 February 1968.

8. Carousel IV Inertial Navigation System (System Technical Description), (no author named), Delco Electronics, Milwaukee, Wisc., September 1971.

9. Kaufman, J., Carousel IV Navigation Computer Equations, XDE 60-S-40, Revision D, AC Electronics (Delco), Milwaukee, Wisc., 24 July 1969.

10. Marnie, J., Documentation of the SIGN III Flight Test Computer Program, Honeywell, Inc., St. Petersburg, Fla., (Internal memorandum), 13 December 1968.

REFERENCES (Cont.)

VOLUME II

11. Gaines, H. T., Description of the Doppler/Inertial/Checkout Kalman Filter for the H-429 Flight Test, Honeywell, Inc., St. Petersburg, Fla., (Internal Report), 24 March 1971.

I2. Musoff, H., SIRU Utilization, Volume I, Theory, Development and Test Evaluation, R-747, C. S. Draper Lab, Cambridge, Mass., March 1974.

13. Oehrle, J. and Whittredge, R., SIRU Utilization, Volume II, Software Description and Program Documentation, R-747, C. S. Draper Lab., Cambridge, Mass., June 1973.

14. Schamp, T., Earth Ellipticity and SIRU Navigation, C. S. Draper Lab., Cambridge, Mass. (Internal memorandum), 6 March 1974.

15. Britting, K. R., Inertial Navigation Systems, Analysis, Wiley-Interscience, New York, N.Y., 1971.

16. McKern, R.A., A Study of Transformation Algorithms for Use in a Digital Computer, (T-493), Master's Thesis, MIT, Cambridge, Mass., January 1968.

17. L. B. Johnson, J. V. Harrison - Multifunction Inertial Sensing, Advanced Inertial Technologies, Volume II, AFAL-TR-73-124.

18. J.I. Freemen, A-7D Mission Analysis and Avionics Configuration Study, Volume I, Technical Report No. 2-50110/2R-3061,(no agency affiliation specified but it appears to be generated at AFAL) December 1972.

19. Kalman, Filter Description, Technical Manual, Northrop Corporation (22915), 7 September 1973.

20. North Atlantic Aided Inertial Navigation System Simulation, Volume 1, W. C. Hoffman et al, Aerospace System Inc. AD-770072, July 1973.

21. Internal CSDL Memo Loran C-Bearing Sea Area, J. Sciegienny, August 27, 1969.

REFERENCES (Cont.)

VOLUME II

22. Technical Report on Ear Antenna Mounted Motion Compensation Investigation, Volumes I, II, III, AFAL-TR-75-106

Additional References for Standard Software Study

Downey, J., Computer Program Development Specification for RACCFT Strapdown Computations, Code Identification Number 56232, Revision F, Sperry Gyroscope, Great Neck, N.Y., October 1974.

Higley, R. B. et al, Micro Navigator (Micron), Phase 1B, Volume II, N57A Development Program, Appendixes A. through H, Technical Report, AFAL-TR-74-178, Autonetics, Anaheim, Cal., July 1974.

Higley, R. B., et al, Micro Navigator (Micron) Phase 1B, Volume III, N57A Development Program, Appendixes I through V, Technical Report, AFAL-TR-74-178, Autonetics, Anaheim, Cal., July 1974.

Browne, B. H., SSMD/ESGN Navigation Mode Computer Algorithm Trade-Off Study (U), (Unclassified portion of Confidential report to Sperry SMD), Honeywell, Inc., St. Petersburg, Fla., 25 June 1971.

Chamberlin, L.A., et al, Design of the Core Elements of the Digital Avionics Information System, Volumes I, II and III, Technical Report, AFAL-TR-74-245, Texas Instruments, Inc., Dallas, Texas, October 1974.

Schmidt, G., and McKern, R., EAR Antenna Mounted Motion Compensation Study, P162, Final Design Review Presentation, C. S. Draper Lab. Inc., Cambridge, Mass., April 1975.

Schmidt, G., et al, EAR Antenna Mounted Motion Compensation Study, Volumes I, II, and III (U), (Volume III is Confidential), AFAL-TR-75-106, C. S. Draper Lab. Inc., Cambridge, Mass., July 1975.

Crews, L., "--derivation of the equations for A-7E navigation functions, - - alignment modes, and - - auto-calibrate mode.", Technical Note 404-128, Naval Weapons Center, China Lake, Cal., January 1972.

Digital Guided Weapons Technology, Design Review Number 1, Summary Report, Report No. DGWT 0120-1, Hughes Aircraft Co., M.S.D., Canoga Park, Cal., 25 November 1974.

Digital Guided Weapons Technology, Design Review Number 2, Summary Report, Report No. DGWT 0120-2, Hughes Aircraft Co., M.S.D., Canoga Park, Cal., 20 January 1975.

Goran, D. and Schiro, J., Definition and Simulation Evaluation of Weapon Delivery Functions in an Advanced Flight Data Management System (U), Technical Report AFFDL-TR-71-10, Hughes Aircraft Co., ASD., , Cal., March 1971, Confidential.

_____, Advanced Aircraft Avionics Requirements Study (U), Final Report, Volumes I and II, MDCE0707 (Contract N00600-72-C-0885), McDonnell Douglas Astronautics Co.,-East, St. Louis, Mo., 24 October 1972, Confidential.

Klees, R., et al, Advanced Air-to-Air Gun Fire Control System Definition Study (U), Technical Report AFAL-TR-73-210, Autonetics Div., Rockwell International, Anaheim, Cal., July 1973, Confidential.

Welp, D., et al, A Computer Program for Simulation and Effectiveness Evaluation of Avionics for Military Aircraft (U), Technical Report AFAL-TR-73-44, Volume II, Battelle, Calumbus Laboratories, February 1973, Confidential.

Kozlowski, D., et al, Advanced Tactical Fighter Target Acquisition and Weapon Delivery Study (U), Volumes I, II, and III, Technical Report ASD/XR-74-21, McDonnell Aircraft Co., M.D.C., St. Louis, Mo., October 1974, Secret.

Levine, S., Minimization of Gravity Uncertainty Errors in Airborne Inertial Navigation Systems (U), TASC Report No. TR-128-1, The Analytic Science Corp., Lowell, Mass., 31 December 1967, Confidential.

Talbot, D., et al, Optimization of Inertial Navigation System Computers by Software Analysis (U), Technical Report AFAL-TR-68-123, Honeywell, Inc., Aerospace Div., St. Petersburg, Fla., July 1968, Confidential.

Rusler, G. et al, Design Configuration and Error Analysis Study of an Airborne Gravimetric System (U), Volume I, Basic Document, Technical Report, AFSWC-TR-71-5, Honeywell, Inc., S&RD, Minneapolis, Minn., December 1970, Confidential.

Rusler, G. et al, Design Configuration and Error Analysis Study of an Airborne Gravimetric System (U), Volume II, Appendix I, Technical Report, AFSWC-TR-71-5, Honeywell, Inc., S&RD, Minneapolis, Minn., December 1970, Confidential.

Rusler, G. et al, Design Configuration and Error Analysis Study of an Airborne Gravimetric System, Volume III, Appendices II through XVI, Technical Report, AFSWC-TR-71-5, Honeywell, Inc., S&RD, Minneapolis, Minn., December 1970.

Higley, R. et al, Micro Navigator (Micron) Phase 1B, Volume I, N57A Development Program, Technical Report, AFAL-TR-74-178, Autonetics Group, Rockwell International, Anaheim, Calif., July 1974.

Rusler, G. et al, Design Configuration and Error Analysis Study of an Airborne Gravimetric System (U), Volume IV, Appendix XVII, Technical Report, AFSWC-TR-71-5, Honeywell, Inc., S&RD, Minneapolis, Minn., December 1970, Secret.

Rusler, G. et al, Design Configuration and Error Analysis Study of an Airborne Gravimetric System (U), Volume V, Appendices XVIII through XXX, Technical Report, AFSWC-TR-71-5, Honeywell, Inc., S&RD, Minneapolis, Minn., December 1970, Confidential.

APPENDIX A
TRANSFORMATION OF COORDINATES

A.1 Introduction

A simple, compact, unambiguous notation describing transformation from one orthogonal axis frame to another is presented in this section.

The notation used in describing the vectors and the coordinates transformations is based on notation used by Britting[15] and the definition of axes frames is based on Honeywell conventions used in SPN/GEANS[2]. The definitions and the notations presented in this Appendix form a base for the proposed symbology of frames and transformations described in Section 6.

A.2 Single Axis Rotations

A coordinate frame, i, is defined by specifying three orthogonal unit vectors $\hat{X}_i$, $\hat{Y}_i$, $\hat{Z}_i$, which form a right handed system, i.e.,

$$\hat{X}_i \times \hat{Y}_i = \hat{Z}_i, \quad \hat{Y}_i \times \hat{Z}_i = \hat{X}_i, \quad \hat{Z}_i \times \hat{X}_i = \hat{Y}_i$$

$$\hat{X}_i \cdot \hat{Y}_i = 0, \quad \hat{Y}_i \cdot \hat{Z}_i = 0, \quad \hat{Z}_i \cdot \hat{X}_i = 0$$

where "x" denotes the vector product and "." the scalar product of two vectors.

The "i" is the short hand name of the frame and is one or more numbers or letters or some combination thereof. The practice of referring to the unit vectors defining a frame as $\hat{i}, \hat{j}, \hat{k}$; $\hat{u}, \hat{v}, \hat{w}$; $\hat{M}, \hat{N}, \hat{O}$; $\hat{X}, \hat{Y}, \hat{Z}$; etc. will not be used because there are too many frames and one would soon run out of triples of letters for which the sequence is obvious. Further, the existence of an x-axis, a y-axis, and a z-axis in each frame is indispensable to the concept of single axis rotations.

Consider two Cartesian coordinate frames with a common origin. Denote the first frame by i and the second by j. Then the unit vectors defining the two frames are $\hat{X}_i$, $\hat{Y}_i$, $\hat{Z}_i$, and $\hat{X}_j$, $\hat{Y}_j$, $\hat{Z}_j$. If one axis of one frame coincides with the corresponding axis of the other frame, i.e., if

$\hat{X}_i = \hat{X}_j$ or $\hat{Y}_i = \hat{Y}_j$ or $\hat{Z}_i = \hat{Z}_j$, then the transformation from one frame to the other is at most a single axis rotation. The axis which is common to the two frames is the axis of rotation and the angle between the corresponding non-common axes is the angle of rotation or argument. The sign of the angle of rotation is positive if the non-common axes of the first ("from") frame move in a clockwise direction to bring themselves into coincidence with the corresponding axes of the second ("to") frame, looking in the positive direction along the common axis (right hand screw rule).

X-axis Rotation, X (α)

If $\hat{X}_i = \hat{X}_j$, the transformation is an x-axis rotation; and if the angle between $\hat{Y}_i$ and $\hat{Y}_j$, and $\hat{Z}_i$ and $\hat{Z}_j$, is α and if α is positive, then transformation from i to j is written as $X(\alpha)$. This is illustrated in Figure A-1.

The equations relating the unit vectors of the j frame axes to the unit vectors of the i frame axes are:

$$\hat{X}_j = \hat{X}_i$$
$$\hat{Y}_j = \cos\alpha\, \hat{Y}_i + \sin\alpha\, \hat{Z}_i$$
$$\hat{Z}_j = -\sin\alpha\, \hat{Y}_i + \cos\alpha\, \hat{Z}_i$$

or in matrix form

$$\begin{bmatrix} \hat{X}_j \\ \hat{Y}_j \\ \hat{Z}_j \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ \hat{Y}_i \\ \hat{Z}_i \end{bmatrix}$$

or

$$X(\alpha) \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

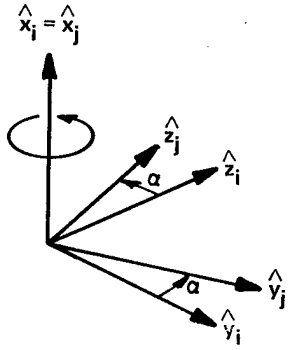


Figure A-1. Rotation About X-axis

In this particular case, the rotation from i to j, denoted by C_i^j is $X(\alpha)$. It is often helpful to represent a number of single axis rotations in a signal flow graph form as shown in Figure A-2.

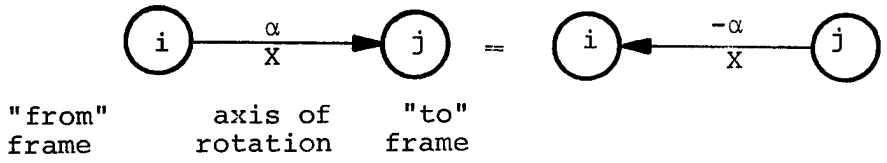


Figure A-2. Rotation About X-axis Flow Diagram

The arrowhead indicates the "to" frame. Changing the direction of the arrow corresponds to changing the sign of the argument or transposing the rotation matrix.

Y-axis Rotation, Y(β)

Simlarly, if $\hat{Y}_j = \hat{Y}_k$, the transformation from the j frame to the k frame is a Y-axis rotation. For a positive argument, β, this is written as Y(β) and is illustrated in Figure A-3.

$$Y(\beta) \triangleq \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix}$$

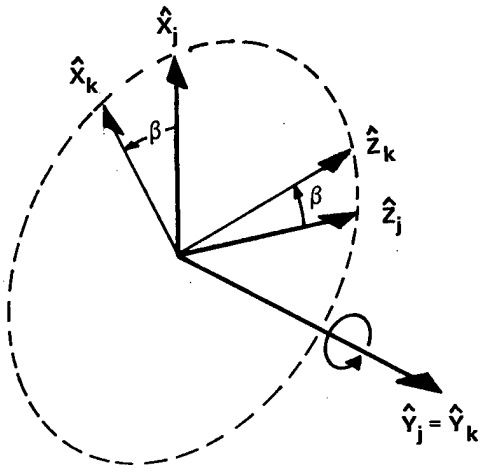


Figure A-3. Rotation about Y-axis

In signal flow form we have for C_j^k (Figure A-4)

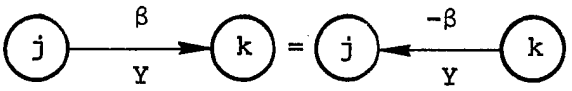


Figure A-4. Rotation About Y-axis Flow Diagram

Z-axis Rotation, Z(γ)

Finally, if $\hat{z}_k = \hat{z}_\ell$, the transformation from the k frame to the ℓ frame, C_k^ℓ, for a positive argument, γ, is written as Z(γ) and is illustrated in Figure A-5.

$$Z(\gamma) \triangleq \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

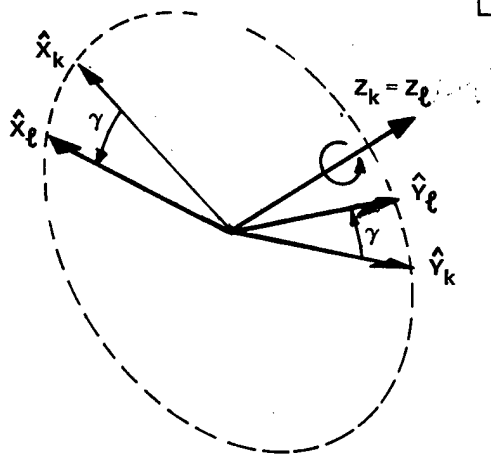


Figure A-5. Rotation About Z-axis

In signal flow form we have for C_k^{ℓ}

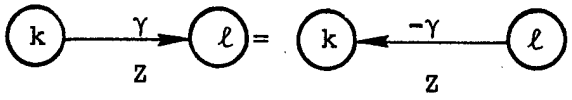


Figure A-6. Rotation about Z-axis Flow Diagram

A.3 The General Rotation

If the three single axis rotations are cascaded, the transformation from the i frame to the ℓ frame, C_i^{ℓ}, may be written as

$$C_i^{\ell} = C_k^{\ell}\, C_j^{k}\, C_i^{j}$$

$$= Z(\gamma)\ Y(\beta)\ X(\alpha)$$

Note, that the subscripts cancel the superscripts from lower left to upper right, leaving only the lower right subscript ("from") and the upper left superscript ("to"). This latter notation appears in Britting[15]. The vector is designated by a small letter with one or more subscripts, if required. The frame in which a vector is defined is denoted by a superscript and transforming a vector $\underline{v}$ from the i frame to the ℓ frame is expressed as

$$\underline{v}^{\ell} = C_i^{\ell}\, \underline{v}^{i}$$

For a vector with notation employing two subscripts, such as the angular velocity of the i^{th} frame with respect to the j^{th} frame, expressed in the k^{th} frame is expressed by $\underline{\omega}^{k}_{j,i}$.

The transformation, C^{ℓ}_{i} or any other sequence of three successive rotations wherein the axis of the middle rotation is different from the axes of rotation on the outside, is often loosely referred to as the Eulerian angle method of specifying the transformation relating any two Cartesian coordinate frames with common origin -- i.e., the Euler angle method permits the specifying the transformation from one orthogonal frame to another by stating the sign and magnitude of the three angles but implies a particular order of the axes of the successive rotations, generally X, Y, Z.

Several of these general rotation matrices occur in the mechanization of navigation equations for the various types of INS. Two examples of frame transformations employed in the navigation computations of the INS with the local-vertical wander-azimuth (LVWA) and with the space-stabilized (SS) platform are shown in Figures A-7 and A-8. In either case, four transformations are required if attitude is computed. Two of the frames indicated, the local-vertical north frame and the body frame, are common to both the LVWA and the SS systems (and also the strapdown system). The number of single axis rotations represented by each of the transformations shown varies from one to four or more.

A prerequisite to establishing a simple, consistent set of single axis rotations relating the frames of interest is to define the coordinate axes of one frame and then relate the others to it via the appropriate rotations.

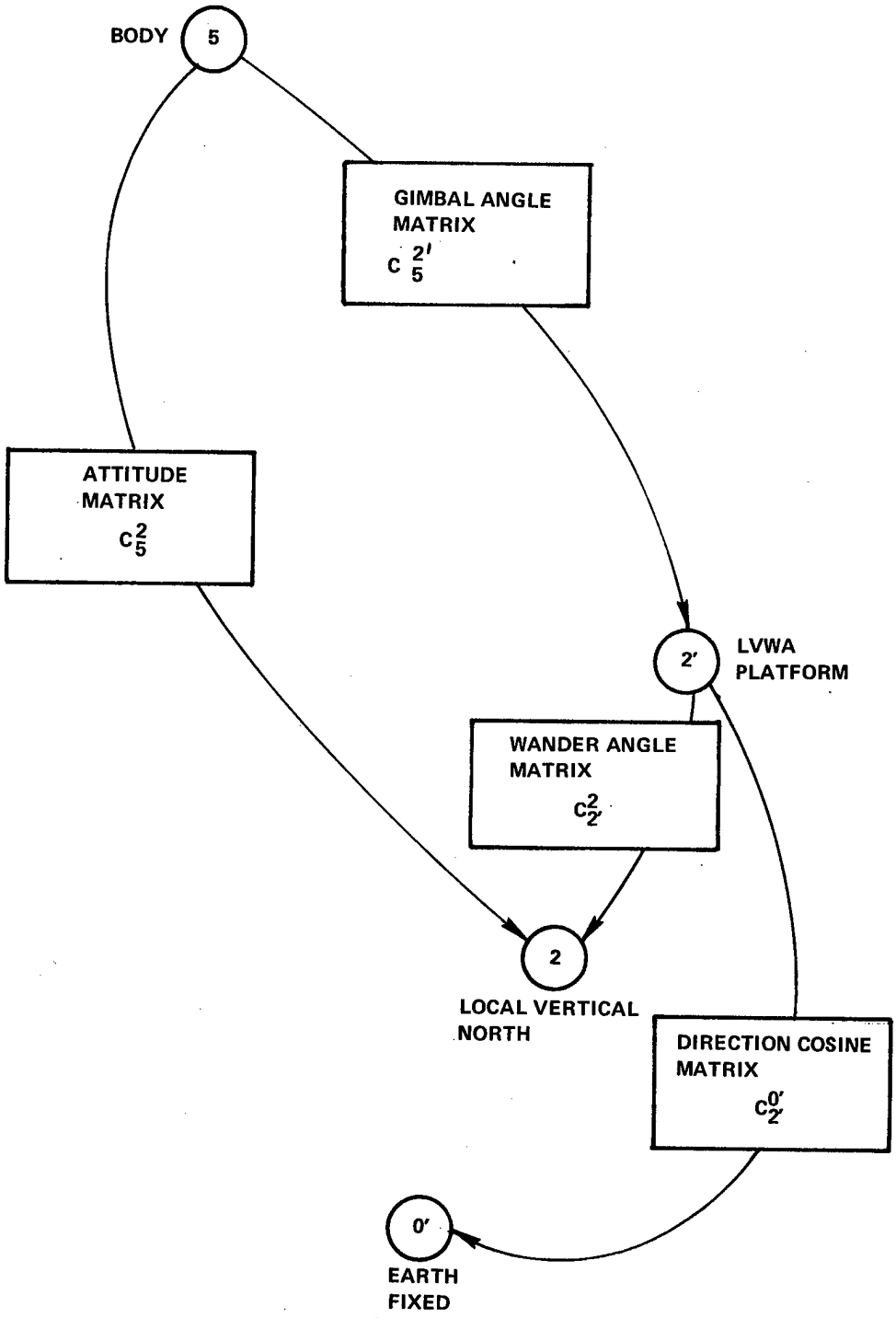


Figure A-7. Transformation Required for Local Vertical Wander-Azimuth INS Navigation Computations in LVWA Computational Frame

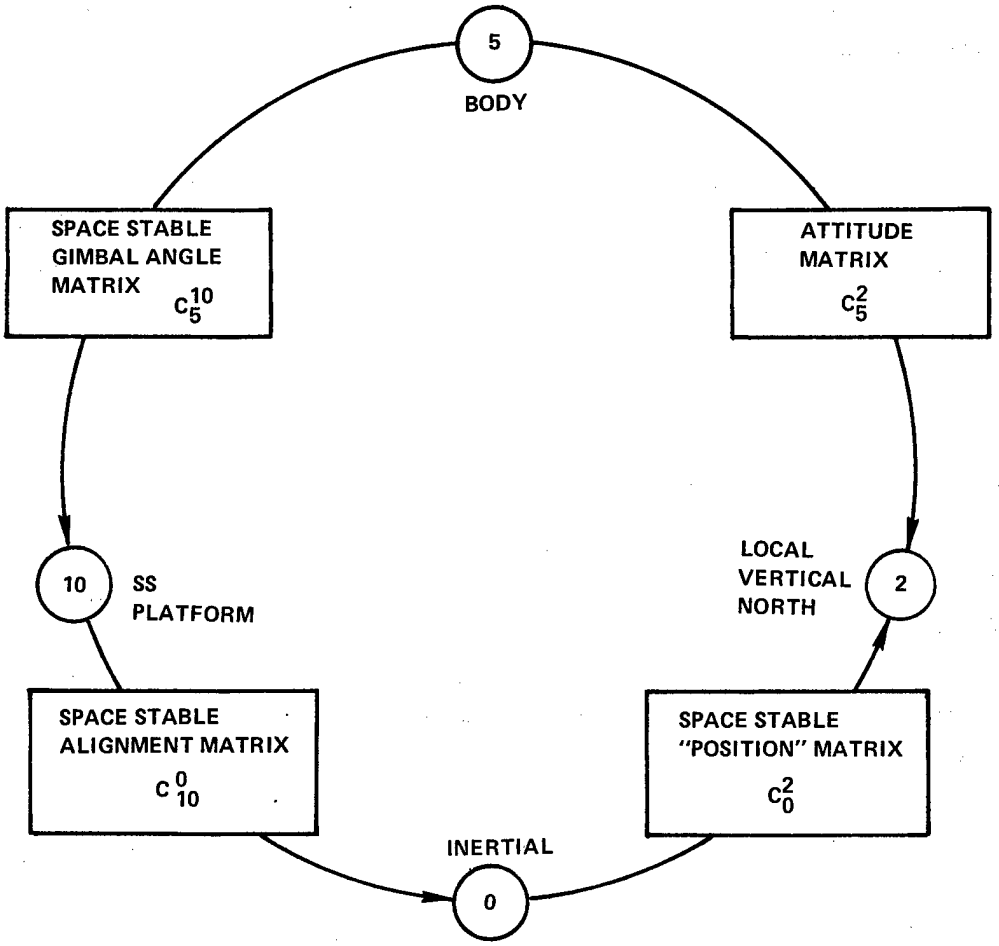


Figure A-8. Transformation Required for Space Stable INS Computation in SS Computational Frame

For simplicity, it is desirable to avoid the introduction of rotations of π or $\pm\pi/2$ which are added to the nominal angles of rotation. This enables one to state that if all the angles relating one frame to another are zero, the corresponding axes of both frames coincide.

Since the choice of the frame to be defined as the starting frame is arbitrary it might as well provide some useful simplification. The one chosen in this discussion simplifies the initialization of a space stable system.

A.4 General Circle Diagram

A.4.1 Introduction

A general circle diagram [2], presenting the coordinate transformations among frames used in the INS navigational computations is shown in Figure A-9 and the angles of rotation are listed in Table A-1. Definition of the coordinates' frames is based on Honeywell conventions used in SPN/GEANS [2] and shown in Figures A-10 and A-11.

The general circle diagram, Figure A-9, was obtained by superimposing the flow graphs of Figures A-7 and A-8 for the LVWA and SS systems and replacing the matrices in the rectangular boxes by appropriate sequences of single axis rotations. Each of the four reference frames fundamental to navigation and attitude computation for the LVWA and SS systems is indicated and named within the rectangular box adjacent to the small circle (node) designating the particular frame. The total of six fundamental frames of interest tabulated in Table A-2 include not only the SS and LVWA systems but also the strapdown (SD) system. (In the latter case, it is assumed that the axes of the strapdown system coincide with the corresponding axes of the body or vehicle frame.)

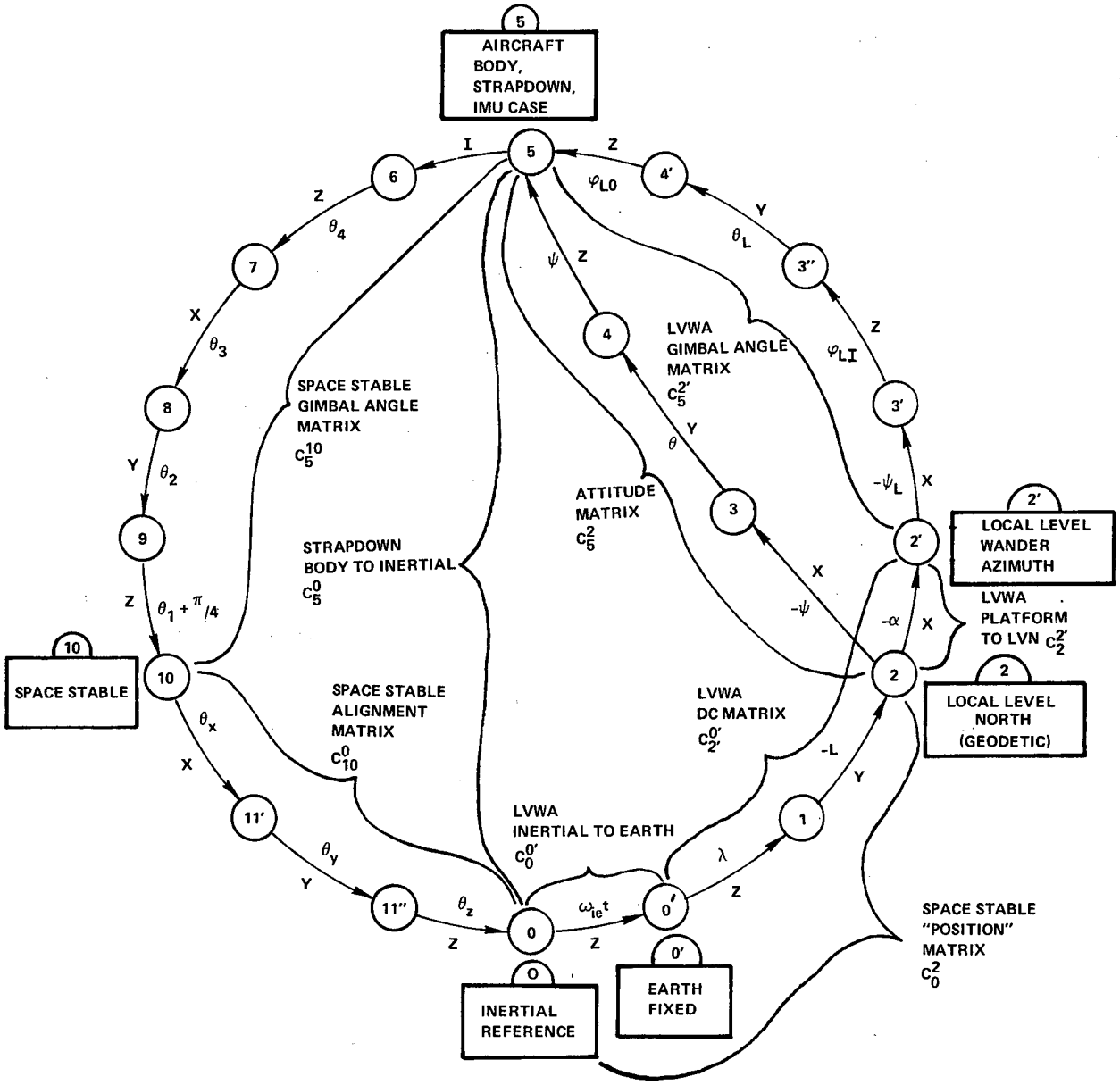


Figure A-9. General Circle Diagram for SS, SD, LVWA INS Computations

Table A-1. Definition of Angles in General Circle Diagram

$\omega_{ie}t$	–	Earth's rate times time (from t = 0)		positive to East
λ	–	Change in longitude (from t = 0)		positive to East
L	–	Geodetic latitude		
ψ	–	Heading (true)		positive to East from North
θ	–	Pitch (true)		positive for nose up
φ	–	Roll (true)		positive for right wing down
α	–	Wander angle		positive to East from North
ψ_L	–	LVWA yaw		positive CW about down (same as ψ)
φ_{LI}	–	LVWA inner roll		(nominally zero) (same as φ)
θ_L	–	LVWA pitch		positive for nose up (same as θ)
φ_{LO}	–	LVWA outer roll		positive for right wing down (same as φ)
θ_4	–	SS gimbal #4		positive for CW rotation of inner gimbal w.r.t. outer gimbal
θ_3	–	SS gimbal #3		
θ_2	–	SS igmbal #2	(nominally zero)	
θ_1	–	SS gimbal #1		
θ_x	–	SS	Euler angles implicit in alignment matrix. Assumed constant in this study.	
θ_y	–	SS		
θ_z	–	SS		

Euler angles for Strapdown alignment matrix not shown since they are not found explicitly.

Table A-2. Coordinate Frames of Interest for SS, SD, and LVWA Systems with SS or LVWA Navigation

The six coordinate frames of interest for any of the six cases indicated are:

0 –frame	THE INERTIAL REFERENCE FRAME defined at moment of entry into navigation i.e. at t = 0.
0′ –frame	THE EARTH FIXED FRAME coincides with 0–frame at t = 0.
2–frame	THE LOCAL VERTICAL NORTH FRAME is the frame of the position and velocity outputs.
2′–frame	THE LOCAL VERTICAL WANDER AZIMUTH FRAME coincides with stable element of LVWA platform and generally with 2–frame at t = 0.
5–frame	THE BODY OR STRAPDOWN PLATFORM FRAME defined by lateral and longitudinal axes of the vehicle.
10–frame	THE SPACE STABLE PLATFORM FRAME coincides with stable element of SS platform.

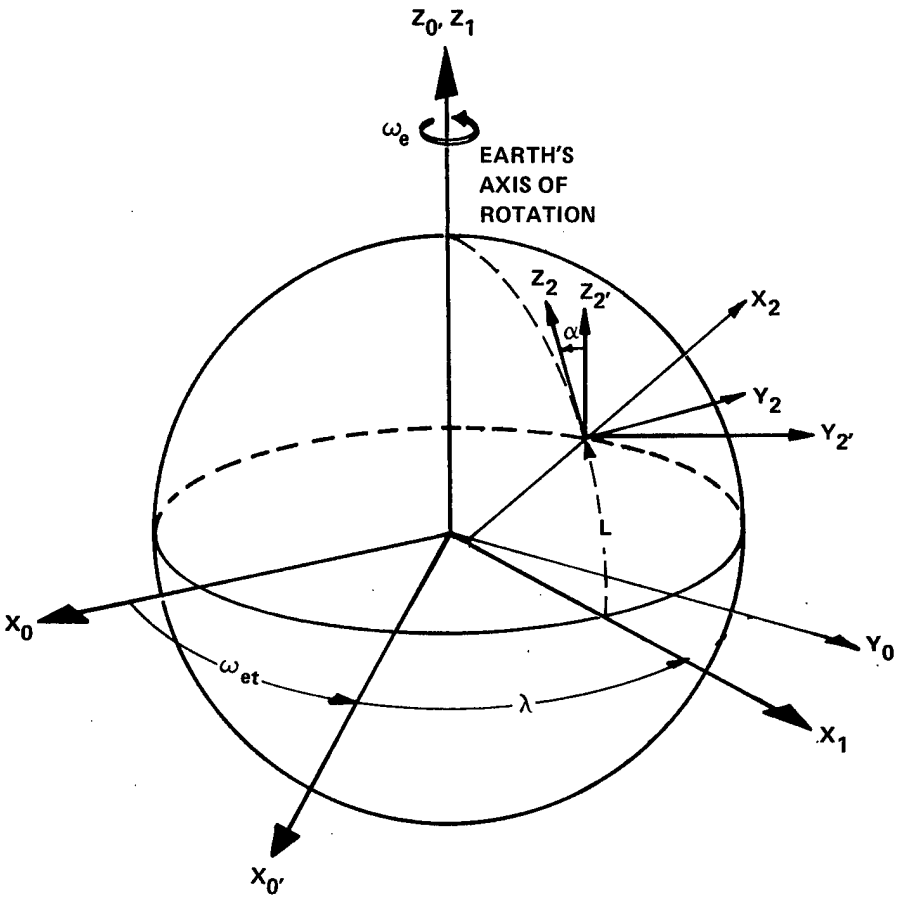


Figure A-10. Coordinate Frames Used in SPN/GEANS Computations

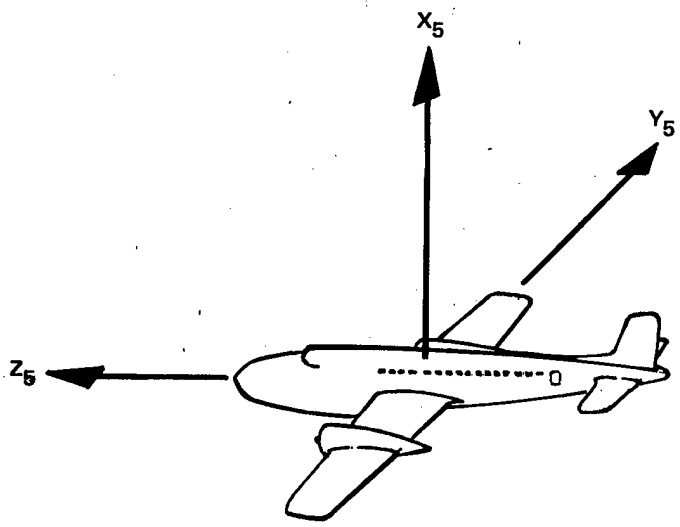


Figure A-11. Vehicle Frame Axes Used in SPN/GEANS Computations

Each individual node (result of a single axis rotation) is specified by a numeral, starting with 0 (zero) for the inertial reference frame. Closely related frames may be designated by the same numeral with varying numbers of superscripted primes (').

A.4.2 Inertial Reference Frame

The definition of the inertial reference frame is illustrated in Figure A-10. Under these definitions, the transformations from the local vertical north frame ,2, to the inertial frame at t = 0 is simply Y(L), where L is the geodetic latitude of the INS navigation position, λ, is the change in longitude from the initial longitude.

A.4.3 The Space Stable "Position" Matrix, C_0^2, or Inertial to Local Vertical North Transformation

This transformation represented by the first three single axis rotations proceeding counterclockwise from the inertial frame, 0, (Figure A-9) is shown in Figure A-12.

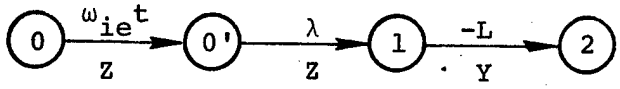


Figure A-12. Inertial-to-Local Vertical Transformation

It is written as

$$C_0^2 = Y(-L)\ Z(\lambda)\ Z(\omega_{ie}t)$$

$$= Y(-L)\ Z(\lambda+\omega_{ie}t)$$

$$= \begin{bmatrix} \cos L \cos\Lambda & \cos L \sin\Lambda & \sin L \\ -\sin\Lambda & \cos\Lambda & 0 \\ -\sin L \cos\Lambda & -\sin L \sin\Lambda & \cos L \end{bmatrix}$$

where $\Lambda = \lambda + \omega_{ie}t$ is the change in celestial longitude from $t = 0$ and is given by

$$\Lambda = \tan^{-1}\left(\frac{Y}{X}\right)$$

where X and Y are the equatorial components of inertial position.

The geodetic latitude, L_g is computed from

$$L_g = \tan^{-1}\left[\left(\frac{1}{1-\varepsilon^2}\right)\frac{Z}{\sqrt{X^2 + Y^2}}\right]$$

where Z is the polar component of inertial position and ε is the eccentricity of the meridional ellipse.

For output, the geodetic longitude, λ_g, is computed from

$$\lambda_g = \lambda_{go} + \Lambda - \omega_{ie}t$$

where λ_{go} is the longitude east of Greenwich at $t = 0$.

In the space stabilized navigation computations, C_0^2 is employed in the transformation of inertial velocity to the local vertical north (LVN) frame, in the attitude computations and, in aided systems, in the application of velocity damping and position reset calculations.

A.4.4 The Attitude Matrix, C_5^2, or Body to Local Vertical North Transformation

The transpose of this transformation, C_2^5, may be represented by the three single axis rotations proceeding from the terminus of the preceding transformation (the LVN frame or 2-frame) to the body frame or 5-frame. Alternate methods of representing the same transformations follow the more complex routes through the space stable platform frame ,0, or the local vertical wander azimuth frame, 2,. A third

alternate representation of the attitude matrix is indicated as embodying the (strapdown) body to inertial frame transformation and the "position" matrix. All routes between two nodes must yield the same transformation. In the signal flow form of Figure A-9, C_2^5, is shown in Figure A-13.

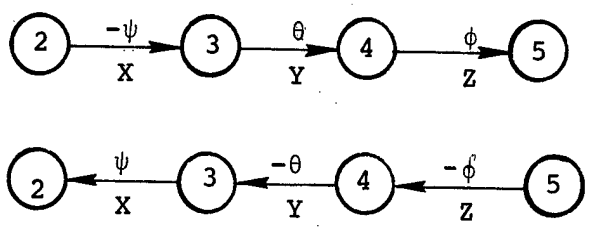


Figure A-13. Body to Local Vertical Transformations

It is written as

$$C_2^5 = Z(\phi)\ Y(\theta)\ X(-\psi)$$

The attitude angles, ϕ, θ, ψ, correspond to the gimbal angles of a perfect, local geodetic vertical, north-slaved, three gimbal platform, with a gimbal order of azimuth (up), pitch (right), roll (forward). Changes in the attitude angles are all positive as the vehicle goes from straight and level flight into a climbing right hand turn. From the circle diagram, Figure A-9, all the attitude angles are zero when the vehicle is level and heading north. The attitude matrix is given by

$$C_5^2 = X(\psi)\ Y(-\theta)\ Z(-\phi) \overset{\Delta}{=} [a_{ij}]$$

$$= \left[\begin{array}{lll|l} \cos\theta\cos\phi & & & -\cos\theta\sin\phi & \sin\theta \\ \cos\psi\sin\phi - \sin\psi\sin\theta\cos\phi & & & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \sin\psi\cos\theta \\ -\sin\psi\sin\phi - \cos\psi\sin\theta\cos\phi & & & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\theta \end{array}\right]$$

and the attitude angles are:

$$\psi = \tan^{-1}\left(\frac{a_{23}}{a_{33}}\right)$$

$$\theta = \tan^{-1}\left(\frac{a_{13}}{\sqrt{a_{11}^2 + a_{22}^2}}\right) = \tan^{-1}\left(\frac{a_{13}}{\sqrt{a_{23}^2 + a_{33}^2}}\right)$$

$$\phi = \tan^{-1}\left(\frac{-a_{12}}{a_{11}}\right)$$

The axes of rotation are permuted whenever the definition of the inertial and hence the LVN frame is changed; however, all attitude matrices based on the same definition of the attitude angles are related by similarity transformations.

The attitude matrix must be computed for a space stable system, and may be computed as part of the high speed computations for a north-slaved, geodetic vertical strapdown system. The three gimbal, horth-slaved, geodetic vertical system provides attitude directly. A four gimbal, LVWA system--in a high performance aircraft, where the inner roll angle, ϕ_{LI}, is not always zero must also compute the attitude matrix.

A.4.5 The LVWA "Direction Cosine "Matrix, $C_{2'}^{0'}$, or LVWA Platform to Earth Fixed Transformation

The "direction cosine" matrix for the local vertical wander azimuth (LVWA) system is the transformation from the LVWA frame, 2', to the earth fixed frame, 0', $C_{2'}^{0'}$. Of course, any rotation is a direction cosine matrix but only the one matrix (and most of its elements individually) appear explicitly in most mechanizations of LVWA systems.

In terms of the nomenclature of Figure A-9, the LVWA platform to the earth fixed frame transformation, $C_{2'}^{0'}$ is shown in signal flow form in Figure A-14.

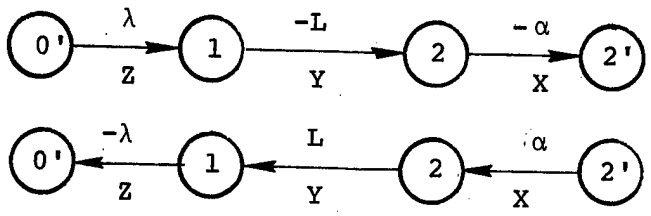


Figure A-14. LVWA to Earth Fixed Frame Transformation

It is written as:

$$C_{2'}^{0'} = Z(-\lambda)\ Y(L)\ X(\alpha) \overset{\Delta}{=} [A_{ij}]$$

$$= \left[\begin{array}{l|l|l} \cos\lambda\ \cos L & \cos\lambda\ \sin L\ \sin\alpha - \sin\lambda\ \cos\alpha & -\cos\lambda\ \sin L\ \cos\alpha - \sin\lambda\ \sin\alpha \\ \sin\lambda\ \cos L & \sin\lambda\ \sin L\ \sin\alpha + \cos\lambda\ \cos\alpha & -\sin\lambda\ \sin L\ \cos\alpha + \cos\lambda\ \sin\alpha \\ \sin L & -\cos L\ \sin\alpha & \cos L\ \cos\alpha \end{array}\right]$$

The elements of the third row of $C_{2'}^{0'}$, are used in the computations of angular velocities and gravity. In addition the first column of $C_{2'}^{0'}$, also enters into the position and wander angle computations viz:

$$L = \tan^{-1}\left(\frac{A_{31}}{\sqrt{A_{11}^{\ 2} + A_{21}^{\ 2}}}\right) = \tan^{-1}\left(\frac{A_{31}}{\sqrt{A_{32}^{\ 2} + A_{33}^{\ 2}}}\right)$$

$$\lambda_g = \tan^{-1}\left(\frac{A_{21}}{A_{11}}\right) + \lambda_{go}$$

$$\alpha = \tan^{-1}\left(\frac{-A_{32}}{A_{33}}\right)$$

Note that longitude, λ_g, and wander angle, α, are indeterminate at the poles ($L = \pm\pi/2$). Some special logic will be required at this point so that $\tan^{-1}$ (0/0) will return the previous value of λ_g or α.

A.4.6 The Space Stable Gimbal Angle Matrix, C_5^{10} or Body to Space Stable Platform Transformation

Four single axis rotations are normally required for this transformation. The trivial, in this case, identity matrix, I, relating the body frame, 5, to the platform case frame, 6, is normally reserved for misalignments of the case with respect to the vehicle. In a non-trivial case C_5^{10} could accommodate an orientation of the outer gimbal along the yaw axis rather than the roll axis of the vehicle.

The signal flow form of C_5^{10} is shown in Figure A-15.

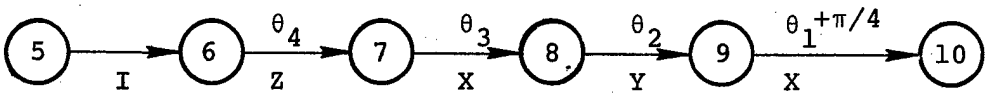


Figure A-15. Body to Space Stable Frame Transformation

Calling $\theta_1 + \pi/4 \rightarrow \theta'_1$

C_5^{10} is written as

$$C_5^{10} = Z(\theta'_1)\ Y(\theta_2)\ X(\theta_3)\ Z(\theta_4)\ I$$

For reasonable computational efficiency and memory utilization, C_5^{10} is subdivided into two two axis rotations viz:

$$C_5^{10} = C_8^{10}\ C_5^8$$

where

$$C_8^{10} = Z(\theta'_1)\ Y(\theta_2)$$

$$= \begin{bmatrix} \cos\theta_1' \cos\theta_2 & \sin\theta_1' & -\cos\theta_1 \sin\theta_2 \\ -\sin\theta_1' \cos\theta_2 & \cos\theta_1' & \sin\theta_1' \sin\theta_2 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix}$$

and

$$C_5^8 = X(\theta_3)\ Z(\theta_4)$$

$$= \begin{bmatrix} \cos\theta_4 & \sin\theta_4 & 0 \\ -\cos\theta_3 \sin\theta_4 & \cos\theta_3 \cos\theta_4 & \sin\theta_3 \\ \sin\theta_3 \sin\theta_4 & -\sin\theta_3 \cos\theta_4 & \cos\theta_3 \end{bmatrix}$$

During navigation, C_5^{10} is employed solely for attitude computation.

A.4.7 The LVWA Gimbal Angle Matrix, $C_5^{2'}$ or Body to LVWA Platform Transformation

This transformation performs the same function in a four gimbal, LVWA platform mechanization as C_5^{10} does in a SS platform mechanization.

However, assuming that the gimbal order for the LVWA platform is azimuth, inner roll, pitch, outer roll, the order of the single axis rotations is different from that specified for the particular SS system (GEANS) shown.

The signal flow form is shown in Figure A-16:

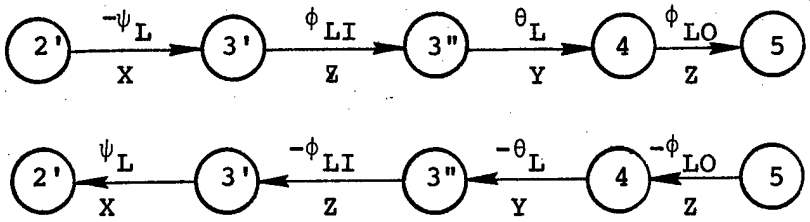


Figure A-16. Body to LVWA Platform Frame Transformations

$C_5^{2'}$ is written as:

$$C_5^{2'} = X(\psi_L)\ Z(-\phi_{LI})\ Y(-\theta_L)\ Z(-\phi_{LO})$$

For the same reasons as before, $C_5^{2'}$ is subdivided into two two axis rotations

$$C_5^{2'} = C_{3"}^{2'}\ C_5^{3"}$$

where $C_{3"}^{2'} = X(\psi_L)\ Z(-\phi_{LI})$

$$= \begin{bmatrix} \cos\phi_{LI} & -\sin\phi_{LI} & 0 \\ \cos\psi_L \sin\phi_{LI} & \cos\psi_L \cos\phi_{LI} & \sin\psi_L \\ -\sin\psi_L \sin\phi_{LI} & -\sin\psi_L \cos\phi_{LI} & \cos\psi_L \end{bmatrix}$$

and $C_5^{3"} = Y(-\theta_L)\ Z(-\phi_{LO})$

$$= \begin{bmatrix} \cos\theta_L \cos\phi_{LO} & -\cos\theta_L \sin\phi_{LO} & \sin\theta_L \\ \sin\phi_{LO} & \cos\phi_{LO} & 0 \\ -\sin\theta_L \cos\phi_{LO} & \sin\theta_L \sin\phi_{LO} & \cos\theta_L \end{bmatrix}$$

It should be noted that none of the local vertical systems surveyed (and a few contained four gimbal platforms) computed attitude at all. The attitude was obtained directly from the readout of the gimbal resolvers. The assumptions implicit in this are $\phi_{LI} = 0$, therefore (3") = (3'), then (4') = (4) and (3') = (3) and hence $\phi_{LO} = \phi$ and $\theta_L = \theta$.

Finally, $\psi_L + \alpha = \psi$.

A.4.8 Other Transformations

Several other transformations occur in one or more mechanizations. Two are single axis rotations; the wander angle matrix, $C_{2'}^{2} = X(\alpha)$ and the inertial to earth fixed rotation $C_{0}^{0'} = Z(\omega_{ie}t)$; hence they need no further explanation. A third, C_{10}^{0}, is shown in the circle diagram as an Euler angle matrix

$$C_{10}^{0} \triangleq Z(\theta_Z)\ Y(\theta_Y)\ X(\theta_X)$$

However, this definition, i.e., of the rotations is arbitrary and, for the purposes of this study, is constant.

A.4.9 A Computational Simplification

An alternate expression for the LVWA gimbal angle matrix, $C_{5'}^{2'}$, is from Figure A-9.

$$C_{5}^{2'} = X(-\alpha)\ X(\psi)\ Y(-\theta)\ Z(-\phi)$$

$$= X(\psi-\alpha)Y(-\theta)\ Z(-\phi)$$

If the "attitude-like" angles are extracted from $C_{5}^{2'}$ as previously indicated, pitch and roll, θ and ϕ, will be correct and true heading will be obtained simply by adding the wander angle, α, to the "heading-like" angle, $\psi - \alpha$.

The above simplification is used in the attitude computations for all three types of INS employing LVWA computational frame and for the LVWA INS employing SS computational frame. The computational flow diagrams are shown in Section 2.

Appendix B

Derivation of Fundamental Equation of Inertial Navigation

Preliminaries (in Euclidean 3 space)

Angular Velocity (as Vector) is denoted by $\underline{\omega}^{c}_{ba}$, where a,b, and c, are the designators for three different Cartesian coordinate frames. $\underline{\omega}^{c}_{ba}$ is read "the angular velocity of the a frame with respect to the b-frame expressed in c-frame coordinates."

Transformation of a Vector

Angular velocity or any other vector quantity may be expressed in terms of its components in any other Cartesian frame by pre-multiplying the vector by the rotation matrix relating the frame in which the vector is expressed to the frame in which it is desired to express it.

Using the example of the angular velocity vector, $\underline{\omega}^{c}_{ba}$, previously defined, it might desired to express this vector in terms of the coordinates of the a-frame rather than the c-frame. If the rotation from the c-frame to the a-frame, C^{a}_{c}, is known, then

$$\underline{\omega}^{a}_{ba} = C^{a}_{c}\underline{\omega}^{c}_{ba}$$

The Cross Product Operator

In an Euclidean 3 space, there is a one-to-one correspondence between the three elements of a vector and the non-zero elements of a three by three, skew symmetric matrix (isomorphism), i.e.

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \leftrightarrow \begin{bmatrix} 0 & -V_z & V_y \\ V_z & 0 & -V_x \\ -V_y & V_x & 0 \end{bmatrix}$$

In case $\underline{v}$ is an angular velocity, $\underline{\omega}$, the corresponding skew symmetric matrix is denoted by Ω e.g.

$$\underline{\omega}^{c}_{ba} \leftrightarrow \Omega^{c}_{ba}$$

When the Ω precedes a vector, say, $\underline{r}$, then

$$\Omega\underline{r} = \underline{\omega}x\underline{r}$$

Or introducing appropriate subscripts and superscripts

$$\Omega^{c}_{ba}\underline{r}^{c} = \underline{\omega}^{c}_{ba}x\underline{r}^{c}$$

Therefore Ω is called the "cross-product operator". Transforming the cross product operator involves premultiplying Ω by a rotation matrix and post multiplying by the transpose of the rotation matrix.

For the vector form we had (as before)

$$\underline{\omega}^{a}_{ba} = C^{a}_{c}\underline{\omega}^{c}_{ba}$$

For the corresponding skew symmetric form we have

$$\Omega^{a}_{ba} = C^{a}_{c}\ \Omega^{c}_{ba}\ C^{c}_{a}$$

Time Derivative of a Rotation Matrix

The time rate of change of the elements of a rotation matrix in terms of the angular rates expressed in either of the terminal frames is frequently required. For example, given the rotation matrix, C^{a}_{c}, what is $\dot{C}^{a}_{c}$, where the superscript dot denotes ordinary differentiation with respect to time? As may be demonstrated in a number of ways

$$\dot{C}^{a}_{c} = C^{a}_{c}\ \Omega^{c}_{ac} = \Omega^{a}_{ac}\ C^{a}_{c}$$

Derivation

It is now intended to derive the vector expressions for the derivative of the velocity of the platform frame with respect to the earth fixed frame expressed in platform frame coordinates.

First, let $\underline{r}^i$ be the position vector from the center of the earth to the vehicle in Cartesian inertial frame (i) coordinates. The same vector expressed in terms of the coordinates of an earth-fixed, earth centered (e) frame is $\underline{r}^e$.

where
$$\underline{r}^i = C_e^i \underline{r}^e \tag{1}$$

and the angular velocity of the e-frame relative to the i frame is a constant, ω_{ie}. Differentiating (1) with respect to time we have

$$\dot{\underline{r}}^i = C_e^i \dot{\underline{r}}^e + \dot{C}_e^i \underline{r}^e \tag{2}$$

but
$$\dot{C}_e^i = \Omega_{ie}^i C_e^i \tag{3}$$

$$\dot{\underline{r}}^i = C_e^i \dot{\underline{r}}^e + \Omega_{ie}^i C_e^i \underline{r}^e \tag{4}$$

Let
$$\underline{v}^i = C_e^i \dot{\underline{r}}^e \tag{5}$$

and substituting (1) into (4)

$$\dot{\underline{r}}^i = \underline{v}^i + \Omega_{ie}^i \underline{r}^i \tag{6}$$

Differentiating (6) with respect to time we have

$$\ddot{\underline{r}}^i = \dot{\underline{v}}^i + \Omega_{ie}^i \dot{\underline{r}}^i + \dot{\Omega}_{ie}^i \underline{r}^i \tag{7}$$

But noting that $\dot{\Omega}_{ie}^i = 0$, since ω_{ie} is a constant, and substituting (6) into (7) we have.

$$\ddot{\underline{r}}^i = \dot{\underline{v}}^i + \Omega_{ie}^i \underline{v}^i + \Omega_{ie}^i \Omega_{ie}^i \underline{r}^i \tag{8}$$

where $\ddot{\underline{r}}^i$ is the total inertial acceleration expressed in inertial coordinates. In the computational (C) frame (8) becomes

$$C_i^c \ddot{\underline{r}}^i = C_i^c \dot{\underline{v}}^i + (C_i^c \Omega_{ie}^i C_c^i)\ (C_i^c \underline{v}^i) + (C_i^c \Omega_{ie}^i C_c^i)\ (C_i^c \Omega_{ie}^i C_c^i) C_i^c \underline{r}^i \qquad (9)$$

but
$$C_i^c \dot{\underline{v}}^i = \dot{\underline{v}}^c + \Omega_{ic}^c \underline{v}^c \qquad (10)$$

and
$$C_i^c \ddot{\underline{r}}^i = C_i^c (f^i + \underline{G}^c) = \underline{f}^c + \underline{G}^c \qquad (11)$$

where $\underline{f}^i$ is the specific force and $\underline{G}^i$ is mass attraction in the inertial frame. Substituting (10) and (11) into (9).

$$\underline{f}^c + \underline{G}^c = \dot{\underline{v}}^c + \Omega_{ic}^c \underline{v}^c + \Omega_{ie}^c \underline{v}^c + \Omega_{ie}^c \Omega_{ie}^c \underline{r}^c$$

but
$$\underline{g}^c \triangleq \underline{G}^c - \Omega_{ie}^c\ \Omega_{ie}^c \underline{r}^c$$

$$\dot{\underline{v}}^c = \underline{f}^c + \underline{g}^c - (\Omega_{ic}^c + \Omega_{ie}^c)\underline{v}^c$$

but
$$\Omega_{ic}^c = \Omega_{ie}^c + \Omega_{ec}^c$$

$$\boxed{\dot{\underline{v}}^c = f^c + \underline{g}^c - (\Omega_{ec}^c + 2\Omega_{ie}^c)\underline{v}^c}$$

APPENDIX C

THE GEODETIC, THE ASTRONOMIC AND THE GEOCENTRIC LATITUDES

The local mean sea level of the earth is represented by a surface of a geoid. The average surface of geoid is represented by a spheroid (an ellipsoid of revolution about the minor, polar, axis of the ellipse).

The geodetic, the astronomic and the geocentric latitudes definition are shown in Figure C-1. The latitudes are defined by the angles in the local meridian plane. The angles are measured from the line in the equatorial plane, determined by an intersection of the local meridian plane with the equational plane.

The geodetic latitude at the point P is defined by the angle L_g between the line in the equatorial plane and the normal to the surface of the spheroid.

The astronomic latitude of the point P is defined by the angle La between the line in the equatorial plane and the normal to the surface of the geoid.

The geocentric latitude of the point P is defined by the angle L_o between the line in the equatorial plane and the line from the earth center to P.

The difference between the geodetic latitude and the atronomic latitude is caused by the gravity anomalies and it is called deflection of the vertical in the meridian plane. The deflection of the vertical remains in large portion of the earth below 10 arc sec, but 1 arc min deflections are also encountered.

The difference between the geodetic latitude and the geocentric latitude is caused by the earth oblateness. The latitude difference is also effected by the altitude h of the point P. The difference between the geodetic latitude, L_g, and the geocentric latitude, L_o, is given by:

$$L_g - L_o \approx f \left(1 - \frac{h}{r}\right) \sin 2 L_g$$

where f = 1/298 is the earth flattening, R = 6371000 meters is the radius of the earth with equal volume and h is the altitude of point P in meters.

At latitude 45°, near the earth surface, the difference between the geodetic and the geocentric latitudes is approximately 1.6 arc min and the altitude change of 30,000 ft results in 1 arc sec latitude difference.

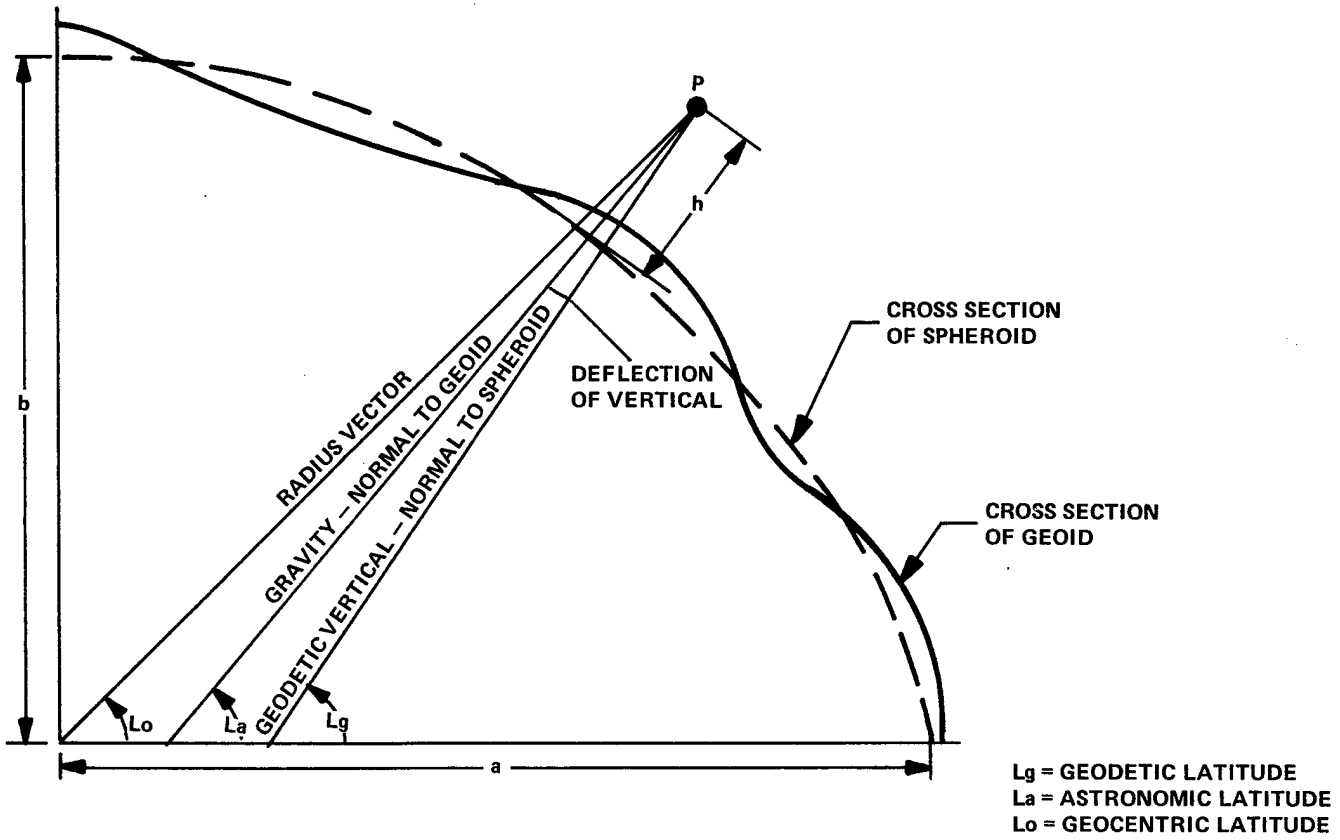


Figure C-1. Latitude Terminology

VOLUME II

APPENDIX D

IMPROVING THE ACCURACY OF THE "BASELINE" LVWA NAVIGATION ALGORITHM

This appendix is addressed to one of the several aspects of the problem, namely, the more accurate numerical integration of the differential equations involved in pure inertial navigation.

The prime inputs to the navigation algorithm, for the present purpose, are the integrals of the components of the specific force in the computational frame, over the navigation computation cycle. Together with the initial conditions, this is all the information required for pure inertial navigation.

The first job of the navigation algorithm, for an earth-relative computational (c) frame, is the integration of the "fundamental equation of inertial navigation", the $\dot{\underline{v}}$ equation from Appendix B, over a navigation computation cycle, Δt. The $\dot{\underline{v}}$ equation is repeated below:

$$\dot{\underline{v}}^c = \underline{f}^c + \underline{g}^c - (\Omega^c_{ec} + 2\,\Omega^c_{ie})\,\underline{v}^c \qquad \text{(D-1)}$$

All the symbols in the above are defined in Appendix B and Section 2.4.1 of this volume. Integrating this equation from some time, t, to $(t + \Delta t)$, we have

$$\int_t^{t+\Delta t} \dot{\underline{v}}^c(\tau)\,d\tau = \int_t^{t+\Delta t} \underline{f}^c(\tau)\,d\tau + \int_t^{t+\Delta t} \underline{g}^c(\tau)\,d\tau$$

$$- \int_t^{t+\Delta t} \left[\Omega^c_{ec}(\tau) + 2\Omega^c_{ie}(\tau)\right]\underline{v}^c(\tau)\,d\tau \qquad \text{(D-2)}$$

Since the object of the integration is to obtain $\underline{v}^c\ (t + \Delta t)$. Equation (D-2) is rearranged as shown below:

$$\underline{v}^c(t+ t) = \underline{v}^c(t) + \underline{\Delta v}^c(t+\Delta t) + \int_t^{t+\Delta t} \underline{g}^c(\tau)\,d\tau$$

$$-\int_t^{t+\Delta t} \Omega_{ec}^c(\tau)\underline{v}^c(\tau)\,d\tau - 2\int_t^{t+\Delta t} \Omega_{ie}^c(\tau)\underline{v}^c(\tau)\,d\tau \qquad \text{(D-3)}$$

where $\underline{\Delta v}^c\ (t + \Delta t) = \int_t^{t+\Delta t} \underline{f}^c\ (\tau)\ d\tau$

Note that $\underline{\Delta v}^c\ (t + \Delta t)$ is the output of an ideal (integrating) accelerometer triad in the computational frame at time $(t + \Delta t)$.

The underlined portions of Equation (D-3) will not change as the algorithm is upgraded; the potential accuracy improvements lie in obtaining better approximations to the three remaining integrals on the right hand side of Equation (D-3).

Before considering the higher order algorithm we note that manner in which the "baseline" algorithm computes the three integrals on the right hand side, viz:

$$\int_t^{t+\Delta t} \underline{g}^c(\tau)\,d\tau \Bigg|_{Base} = \underline{g}^c(t)\Delta t \qquad \text{(D-4)}$$

which assumes $\dot{\underline{g}}^c(t) = 0$

$$-\int_{t}^{t+\Delta t} \Omega_{ec}^{c}(\tau)\underline{v}^{c}(\tau)d\tau \Bigg|_{\text{Base}} = -\Omega_{ec}^{c}(t)\ \underline{v}^{c}(t)\Delta t \tag{D-5}$$

which assumes $\dot{\Omega}_{ec}^{c}(t),\ \dot{\underline{v}}^{c}(t) = 0$

$$-2\int_{t}^{t+\Delta t} \Omega_{ie}^{c}(\tau)\ \underline{v}^{c}(\tau)d\tau \Bigg|_{\text{Base}} = -2\Omega_{ie}^{c}(t)\ \underline{v}^{c}(t)\Delta t \tag{D-6}$$

which assumes $\dot{\Omega}_{ie}^{c}(t),\ \dot{\underline{v}}^{c}(t) = 0$

Hence the baseline navigation algorithm (without vertical damping) is given by

$$\underline{v}^{c}(t+\Delta t)\Big|_{\text{Base}} = \underline{v}^{c}(t)+\Delta\underline{v}^{c}(t+\Delta t) + \underline{g}^{c}(t)\Delta t - \left[\Omega_{ec}^{c}(t) + 2\Omega_{ie}^{c}(t)\right]\ \underline{v}^{c}(t)\Delta t \tag{D-7}$$

There are some computational lags and further approximations in the computation of $\Omega_{ec}^{c}(t)$, the skew symmetric form of $\underline{\rho}^{c}(t)$, but these may be dealt with later.

For notational compactness, it is expedient to define angular velocity vector, $\underline{\omega}^{c}(t)$, and its skew symmetric counterpart, $\Omega^{c}(t)$, as

$$\underline{\omega}^{c}(t) \triangleq \underline{\omega}_{ec}^{c}(t) + 2\underline{\omega}_{ie}^{c}(t) \tag{D-8a}$$

and

$$\Omega^{c}(t) \triangleq \Omega_{ec}^{c}(t) + 2\Omega_{ie}^{c}(t) \tag{D-8b}$$

$$\int_t^{t+\Delta t} \left[\Omega_{ec}^c(\tau) + 2\Omega_{ie}^c(\tau)\right] \underline{v}(\tau)\, d\tau = \int_t^{t+\Delta t} \Omega^c(\tau)\underline{v}^c(\tau)\, d\tau \qquad \text{(D-9)}$$

Expanding the integrands in Eq. (D-3) in Taylor's series about t, we have

$$\underline{\delta v}^c(t+\Delta t)\Big|_{\underline{g}^c} \triangleq \int_t^{t+\Delta t} \underline{g}^c(\tau)\, d\tau \qquad \text{(D-10)}$$

$$= \int_t^{t+\Delta t} \left[\, \underline{g}^c(t) + \dot{\underline{g}}^c(t)(\tau-t) + \ddot{\underline{g}}^c(t)\frac{(\tau-t)^2}{2} + \ldots \right] d\tau$$

$$= \underline{g}^c(t)\Delta t + \dot{\underline{g}}^c(t)\,\frac{\Delta t^2}{2} + \ddot{\underline{g}}^c(t)\,\frac{\Delta t^3}{6} + \ldots$$

Similarly

$$-\underline{\delta v}^c(t+\Delta t)\Big|_{\substack{\Omega^c \\ \underline{v}^c}} \triangleq \int_t^{t+\Delta t} \Omega^c(\tau)\underline{v}^c(\tau)\, d\tau \qquad \text{(D-11)}$$

where $\Omega^c(\tau) = \Omega^c(t) + \dot{\Omega}^c(t)(\tau-t) + \ddot{\Omega}^c(t)\frac{(\tau-t)^2}{2} + \ldots.$

and $\underline{v}^c(\tau) = \underline{v}^c(t) + \dot{\underline{v}}^c(t)(\tau-t) + \ddot{\underline{v}}^c(t)\frac{(\tau-t)^2}{2} + \ldots.$

Then the integrand, $\Omega^c(\tau)\underline{v}^c(\tau)$, is given by

$$\Omega^c(\tau)\underline{v}^c(\tau) = \Omega^c(t)\underline{v}^c(t) \qquad \text{(D-12)}$$

$$+\left[\Omega^c(t)\dot{\underline{v}}^c(t) + \dot{\Omega}^c(t)\underline{v}^c(t)\right](\tau-t) \qquad \text{(D-12)}$$

$$+\left[2\dot{\Omega}^c(t)\dot{\underline{v}}^c(t) + \Omega^c(t)\ddot{\underline{v}}^c(t) + \ddot{\Omega}^c(t)\underline{v}^c(t)\right]\frac{(\tau-t)^2}{2}$$

$$+ \text{terms of the order of } (\tau-t)^3,\ (\tau-t)^4, \text{ etc.}$$

Performing the indicated integration

$$-\delta\underline{v}^c(t+\Delta t)\Big|_{\substack{\Omega^c \\ \underline{v}^c}} = \Omega^c(t)\underline{v}^c(t)\Delta t \qquad \text{(D-13)}$$

$$+\left[\Omega^c(t)\dot{\underline{v}}^c(t) + \dot{\Omega}^c(t)\underline{v}^c(t)\right]\frac{\Delta t^2}{2}$$

$$+\left[\Omega^c(t)\ddot{\underline{v}}^c(t) + 2\dot{\Omega}^c(t)\dot{\underline{v}}^c(t) + \ddot{\Omega}^c(t)\underline{v}^c(t)\right]\frac{\Delta t^3}{6}$$

$$+ \text{ terms of the order of } \Delta t^4 \text{ and higher.}$$

The "baseline" (first order) algorithm may be upgraded to a second order algorithm by adding the second terms (those multiplied by $\frac{\Delta t^2}{2}$) of Equations (D-10) and (D-13) to Equation (D-7).

$$\underline{v}^c(t+\Delta t)\Big|_{\text{order}}^{\text{2nd}} = \underline{v}^c(t) + \underline{v}^c(t+\Delta t) + \underline{g}^c(t)\,\Delta t \qquad \text{(D-14)}$$

$$-\Omega^c(t)\underline{v}^c(t)\Delta t + \dot{\underline{g}}^c(t)\,\frac{\Delta t^2}{2}$$

$$-\left[\Omega^c(t)\dot{\underline{v}}^c(t) + \dot{\Omega}^c(t)\underline{v}^c(t)\right]\frac{\Delta t^2}{2}$$

To generate Equation (D-14), we need the three derivatives (in a vector sense - actually nine derivatives) $\dot{\underline{v}}^c(t)$, $\dot{\underline{g}}^c(t)$ and $\dot{\Omega}^c(t)$ or $\dot{\underline{\omega}}^c(t)$. If these derivatives can be generated, the navigation algorithm can be upgraded not only to the second order but may also include part of the third order algorithm, $\dot{\Omega}^c(t)\dot{\underline{v}}^c(t)\Delta t^3/3$.

The assumptions in the second order algorithm are

$$\ddot{\underline{v}}^c(t),\ \ddot{\underline{g}}^c(t),\ \ddot{\Omega}^c(t) = 0.$$

Subject to the above assumptions, one may regroup the terms in $\underline{g}^c(t)$ and $\dot{\underline{g}}^c(t)$ viz:

$$\underline{\delta v}^c(t+\Delta t)\Big|_{\substack{\underline{g}^c \\ \dot{\underline{g}}^c \\ 2nd}} = \underline{g}^c(t)\Delta t + \dot{\underline{g}}^c(t)\ \frac{\Delta t^2}{2} \qquad \text{(D-15)}$$

$$= \left[\underline{g}^c(t) + \dot{\underline{g}}^c(t)\ \frac{\Delta t}{2}\right]\Delta t$$

$$= \underline{g}^c\left(t+\frac{\Delta t}{2}\right)\Delta t$$

Therefore, if we can obtain an estimate of the extrapolated value of $\underline{g}^c$ at the midpoint point of the next computation cycle, we have effectively incorporated its derivative at the present time. The mechanism for computing the extrapolated value of $\underline{g}^c$, simply consists of noting that

$$\underline{g}^c(t+\frac{\Delta t}{2}) = \underline{f}_1\left[L\left(t+\frac{\Delta t}{2}\right),\ \alpha\left(t+\frac{\Delta t}{2}\right),\ h\left(t+\frac{\Delta t}{2}\right)\right] \qquad \text{(D-16)}$$

where L is geodetic latitude
α is wander angle
h is altitude

as may be seen from any of Equations (2-39) (2-62) or (2-78) in Sections 2.4.1.2, 2.4.3 or 2.4.4 of this volume.

The required functions of L and α are contained in the extrapolated dcm while h may be extrapolated directly. Therefore an estimate of $\underline{g}^c(t+\frac{\Delta t}{2})$ may be obtained at t for use at $(t+\Delta t)$.

Now we attempt to upgrade the $\underline{\delta v}^c(t+\Delta t)\Big|_{\underline{v}^c}^{\Omega^c}$ computation by replacing $\Omega^c(t)\underline{v}^c(t)\Delta t$ by $\Omega^c(t+\frac{\Delta t}{2})\underline{v}^c(t+\frac{\Delta t}{2})\Delta t$

where $\Omega^c(t+\frac{\Delta t}{2}) = \Omega^c(t) + \dot{\Omega}^c(t)\ \frac{\Delta t}{2}$

and $\underline{v}^c(t+\frac{\Delta t}{2}) = \underline{v}^c(t) + \dot{\underline{v}}^c(t)\ \frac{\Delta t}{2}$

if $\ddot{\underline{v}}^c(t),\ \ddot{\Omega}^c(t) = 0$ (D-17)

$$\text{Then } \Omega^c(t+\tfrac{\Delta t}{2})\ \underline{v}^c(t+\tfrac{\Delta t}{2})\ \Delta t = \left[\Omega^c(t)\dot{\Omega}^c(t)\tfrac{\Delta t}{2}\right]\left[\underline{v}^c(t) + \dot{\underline{v}}^c(t)\tfrac{\Delta t}{2}\right]\Delta t$$

$$= \Omega^c(t)\underline{v}^c(t)\Delta t + \left[\Omega^c(t)\underline{v}^c(t) + \dot{\Omega}^c(t)\underline{v}^c(t)\right]\frac{\Delta t^2}{2} + \left[\frac{3}{2}\dot{\Omega}^c(t)\dot{\underline{v}}^c(t)\right]\ \frac{\Delta t^3}{6}$$

Hence using the mid-computation cycle values of angular velocity and velocity not only gives the correct second terms for $(\underline{\omega}x\underline{v})\Delta t$ but also 3/4's of one of the third order terms of Equation (D-13).

The method of obtaining the estimates of $\underline{v}^c(t+\frac{\Delta t}{2})$ and $\Omega^c(t+\frac{\Delta t}{2})$ is straightforward. Before $\underline{v}^c(t)$ is updated at time t, from time $(t-\Delta t)$, the old value is stored and the extrapolated value is estimated as

$$\tilde{\underline{v}}^c(t+\tfrac{\Delta t}{2}) \triangleq \underline{v}^c(t) + \left[\frac{\underline{v}^c(t) - \underline{v}^c(t-\Delta t)}{\Delta t}\right]\frac{\Delta t}{2} \qquad \text{(D-18)}$$

$$= \frac{1}{2}\left[3\underline{v}^c(t) - \underline{v}^c(t-\Delta t)\right]$$

There is, of course, a lag of $\frac{\Delta t}{2}$ in the computed derivative of velocity but it is better than no derivative at all and besides it is very inexpensive computationally.

The estimated angular velocity, $\underline{\omega}^c$, at time $(t+\frac{\Delta t}{2})$ may be written as

$$\Omega^c(t+\tfrac{\Delta t}{2}) = F_2\left[\underline{v}^c(t+\tfrac{\Delta t}{2}),\ L(t+\tfrac{\Delta t}{2}),\ \alpha(t+\tfrac{\Delta t}{2}),\ h(t+\tfrac{\Delta t}{2})\right] \qquad \text{(D-19)}$$

That is, $\underline{\omega}^c(t+\frac{\Delta t}{2})$ is a function of the same variables as $\underline{g}^c(t+\Delta t)$ plus the linear velocity $\underline{v}^c(t+\frac{\Delta t}{2})$, which latter we have just estimated.

The extrapolated dcm, $C_c^e(t+\frac{\Delta t}{2})$, which contains the appropriate functions of L and α, is obtained as an almost painless byproduct of the update of the dcm from time $(t-\Delta t)$ to time t (the second integration). The dcm update, per se, which is assumed to be second or higher order, constructs an additive "update" matrix $\delta C_c^e(t,t-\Delta t)$ which is stored, then added to the "old" dcm, $C_c^e(t-\Delta t)$ as shown below

$$C_c^e(t) = C_c^e(t-\Delta t) + \Delta C_c^e(t,t-\Delta t) \qquad \text{(D-20)}$$

The extrapolated dcm is given by

$$\tilde{C}_c^e(t+\tfrac{\Delta t}{2}) = C_c^e(t) + \tfrac{1}{2}\,\delta C_c^e(t,t-\Delta t) \qquad \text{(D-21)}$$

The "update" matrix itself is of the form

$$\delta C_c^e(t,t-\Delta t) = C_c^e(t-\Delta t)\left\{\Omega_{ec}^c(t-\tfrac{\Delta t}{2})\Delta t + \left[\Omega_{ec}^c(t-\tfrac{\Delta t}{2})\right]^2\frac{\Delta t^2}{2} + \right\} \qquad \text{(D-22)}$$

where $\Omega_{ec}^c(t-\frac{\Delta t}{2})$ is the skew symmetric form of $\underline{\omega}_{ec}^c(t-\frac{\Delta t}{2})$ or $\underline{\rho}^c(t-\frac{\Delta t}{2})$, the angular velocity of the computational frame with respect to the earth fixed frame, expressed in the computational frame, and computed at midpoint of the computation cycle.

The angular velocity, $\Omega^c_{ec}(t-\frac{\Delta t}{2})$, is a function of the same variables as $\Omega^c(t-\frac{\Delta t}{2})$ i.e.,

$$\Omega^c_{ec}(t-\tfrac{\Delta t}{2}) = F_3\left[\underline{v}^c(t-\tfrac{\Delta t}{2}),\ L(t-\tfrac{\Delta t}{2}),\ \alpha(t-\tfrac{\Delta t}{2}),\ h(t-\tfrac{\Delta t}{2})\right] \quad \text{(D-23)}$$

Rather than use the extrapolated value of $\underline{v}^c$ from the previous computation cycle, it is more accurate and almost as simple to use an interpolated value of $\underline{v}^c$ from the present computation cycle.

Define $\underline{v}^c(t-\frac{\Delta t}{2})$ by

$$\underline{v}^c(t-\tfrac{\Delta t}{2}) \triangleq \frac{1}{2}\left[\underline{v}^c(t) + \underline{v}^c(t-\Delta t)\right] \quad \text{(D-24)}$$

For the remaining three variables (or functions thereof) it is sufficient to use the extrapolated values from the previous computation cycle.

The other part of $\Omega^c(t-\frac{\Delta t}{2})$, $\Omega^c_{ie}(t-\frac{\Delta t}{2})$, is strictly a function of L, α, and ω_{ie} i.e.,

$$\Omega^c_{ie}(t-\tfrac{\Delta t}{2}) = \omega_{ie} F_4\left[L(t-\tfrac{\Delta t}{2}),\ \alpha(t-\tfrac{\Delta t}{2})\right] \quad \text{(D-25)}$$

Here, F_4 is obtained directly from the elements of the extrapolated dcm from the previous computation cycle.

Putting all the foregoing together in the appropriate order, along with suitable initialization, and vertical damping (see Appendix F), and we have the "Upgraded LVWA Navigation Algorithm" (see Section 2.4 of this volume) - which might approach the accuracy of a vastly simpler space stable navigation algorithm.

Volume II

APPENDIX E

APPROXIMATING THE MID COMPUTATION CYCLE TRANSFORMATION

In the strapdown computations the mid-computation cycle transformation from the body frame to the computational frame, $C^c_{b_{N-1/2}}$, is approximated by

$$\tilde{C}^c_{b_{N-1/2}} = \frac{1}{2}\left(C^c_{b_N} + C^c_{b_{N-1}}\right)$$

A direction cosine matrix or quaternion update is used to update $C^c_{b_{N-1}}$ to $C^c_{b_N}$. Consider the d.c.m. update. Then

$$C^c_{b_N} = C^c_{b_{N-1}}\, C(\theta)$$

where $C(\theta) = I + R \sin\theta + R^2 (1-\cos\theta)$

and $R = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}$ where a,b,c are the direction cosines of the axis of rotation

Hence $\tilde{C}^c_{b_{N-1/2}} = \frac{1}{2} C^c_{b_{N-1}} [I + I + R \sin\theta + R^2 (1-\cos\theta)]$

$$= C^c_{b_{N-1}} + C^c_{b_{N-1}} \left[R \frac{\sin\theta}{2} + R^2 \left(\frac{1-\cos\theta}{2}\right)\right]$$

The exact expression for $C^c_{b_{N-1/2}}$, assuming $\dot{\theta}$ is a constant, is given by

$$C^c_{b_{N-1/2}} = \left[I + R \sin\left(\frac{\theta}{2}\right) + R^2 \left(1-\cos\left(\frac{\theta}{2}\right)\right)\right]$$

$$= C^c_{b_{N-1}} + C^c_{b_{N-1}} \left[R \sin\left(\frac{\theta}{2}\right) + R^2 \left(1-\cos\left(\frac{\theta}{2}\right)\right)\right]$$

The error is the approximation is

$$\tilde{C}^{c}_{b_{N-1/2}} - C^{c}_{b_{N-1/2}} = C^{c}_{b_{N-1}} \left\{ R \left[\frac{\sin\theta}{2} - \sin\left(\frac{\theta}{2}\right) \right] + R^2 \left[\left(\frac{1-\cos\theta}{2} \right) - \left(1 \cos\left(\frac{\theta}{2}\right) \right) \right] \right\}$$

Expanding sin θ, cos θ, $\sin\left(\frac{\theta}{2}\right)$, $\cos\left(\frac{\theta}{2}\right)$ to the third power in θ

$$\sin\theta = \theta - \frac{\theta^3}{6}, \quad \cos\theta = 1 - \frac{\theta^2}{2}$$

$$\sin\left(\frac{\theta}{2}\right) = \frac{\theta}{2} - \frac{\theta^3}{48}, \quad \cos\left(\frac{\theta}{2}\right) = 1 - \frac{\theta^2}{8}$$

Inserting these expressions and simplifying gives

$$\tilde{C}^{c}_{b_{N-1/2}} - C^{c}_{b_{N-1/2}} = C^{c}_{b_{N-1}} \left[-R \frac{\theta^3}{16} + R^2 \frac{\theta^2}{8} \right]$$

If rotation is entirely about the roll (longitudinal) axis, then the direction cosines of the axis of rotation are {a,b,c} = { 0 0 1}

and $R = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $R^2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

The $-R\theta^3/16$ term will produce a very small rotational error in the normal and lateral components of the integral of specific force which would cancel out at the end of the turn.

The $R^2\ \theta^2/8$ will produce a scale factor error in both the normal and lateral components which will be rectified and hence, will not cancel out at the completion of the turn. The nature of this error is such that it does not propagate directly into an attitude error, but since it introduces a scale factor error into the incremental velocity while the attitude is changing, it will integrate into velocity and hence position error and into the computed vehicle rate where it will ultimately produce an attitude error.

The magnitude of the lateral velocity error introduced during the roll-up and roll down portion of a 1/2-g horizontal turn, when the constant roll rate is 180 degrees per second and the strap-down computation cycle, Δt, is 1/128 of a second is computed below.

The scale factor error, ε_{SF}, is

$$\varepsilon_{SF} = \frac{\dot{\phi}^2 \Delta t^2}{8} = \frac{\pi^2 x (2^{-7})^2}{8} = \pi^2 \times 2^{-7} = 75 \text{ ppm}$$

The maximum roll angle, $\dot{\phi}_m$, is

$$\phi_m = \tan^{-1} 1/2 = 26.656°$$

The roll up time, t_u, is

$$t_u = \frac{\phi_m}{\dot{\phi}} = 0.1475 \text{ sec}$$

The lateral velocity error at the completion of the term is the scale factor error times the area under the lateral acceleration curve

$$\varepsilon V | \varepsilon_{SF} = \varepsilon_{SF} \left(\frac{1}{2} g_{max} 2t_u \right)$$

$$= 7.5 \times 10^{-5} \times 16 \times .1475 = 0.177 \times 10^{-3} \text{ fps}$$

This is a significant error. It may be eliminated in one of two ways. The simplest is to alternate the "attitude" matrix update with the ΔV transformation so that no interpolation is required, which is a practical solution adopted by some strapdown systems. The second, which is applicable to the present case i.e. PROFGEN/ NUMSIM, consists of performing a second update of the "attitude" matrix using half the incremental angles and a second or higher order update.

The efficient incorporation of these operations into the d.c.m. and quaternion updates should be investigated during any follow-on efforts.

VOLUME II

APPENDIX F

VERTICAL DAMPING FOR THE "UPGRADED" ALGORITHM

The assumption invoked throughout the algorithm development has been (and still is) that information is available at the end of a computation cycle and that outputs are to be provided appropriate to the end of the computation cycle.

This implies that the computer time required to update the outputs (which is in the range of a few milliseconds to tens of milliseconds) is a negligible delay. For some purposes this delay may not be acceptable and specific steps may have to be taken to extrapolate INS output parameters ahead to the time of their intended use.

At any rate, there is a premium to be placed on minimizing the delay between accepting input data and outputing the updated navigation and attitude data, particularly the rapidly changing outputs, such as velocity. Therefore, the data required for velocity updating, including the vertical damping, but excepting the incremental velocity, are all computed at the end of the previous computation cycle for use in the early portions of the present update cycle.

The rationale for the improved accuracy navigation algorithm was outlined in Appendix D of this volume, while the algorithm proper was presented in Section 2.4 of this volume. The continuous form of the damped vertical channel was shown in Section 2.4.2 of this volume.

The only additional coding specifically required to upgrade the vertical damping mechanization is the extrapolation of the reference altitude to mid-computation cycle. (The system computed altitude was already extrapolated to mid-computation cycle for the "gravity" computation).

The upgraded vertical channel computations are shown in Figure F-1 in the detailed computation flow diagram. Below the dashed line are the (linearized) undamped vertical damping calculations proper. In Figure F-1, the inputs to the "unit delay" boxes, were computed during the previous cycle, and become the outputs of the boxes during the present cycle (after which the inputs are updated again). It is obvious from the figure that a very modest number of arithmetic (consisting of summation and scaling) operations are required to update the velocity - once the incremental velocity is available.

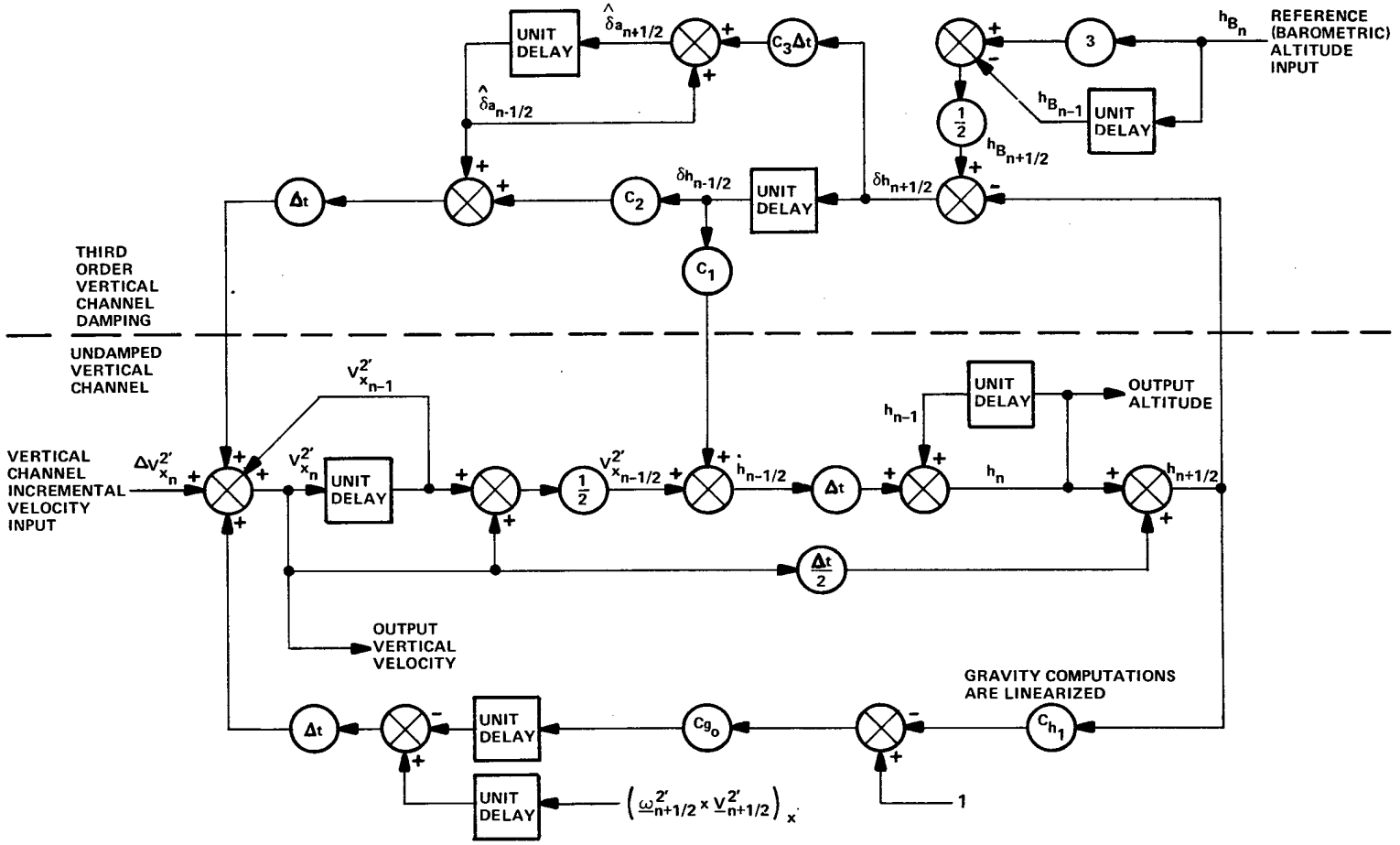


Figure F-1. Damped Vertical Channel Computations for "Upgraded" Algorithm (at end of n^{th} computation cycle)

R-977

VOLUME II

DISTRIBUTION LIST

R. Bumstead
W. Caffery
A. Ciccolo
G. Coate
R. Crisp
K. Daly
M. Dare
W. Delaney
W. Denhard
J. DiSorbo
K. Fertig
J. Fish
J. Harper
R. Harris
J. Hursh
P. Kampion
J. Kishel
W. Koenigsberg
R. Leger
R. Marshall
B. McCoy
W. McFarland
R. Nurse
P. Peck
T. Reed
D. Riegsecker
G. Schmidt (2)
J. Sciegienny (2)
R. Setterlund
L. W. Torrey
R. Wexler
TIC (5)

Air Force Avionics Laboratory (20)
Attn: RWA-3
Wright-Patterson Air Force Base,
Ohio 45433

Air Force Avionics Laboratory (20)
Attn: Capt. E. Harrington, RWA-2
Wright-Patterson Air Force Base,
Ohio 45433